Interface

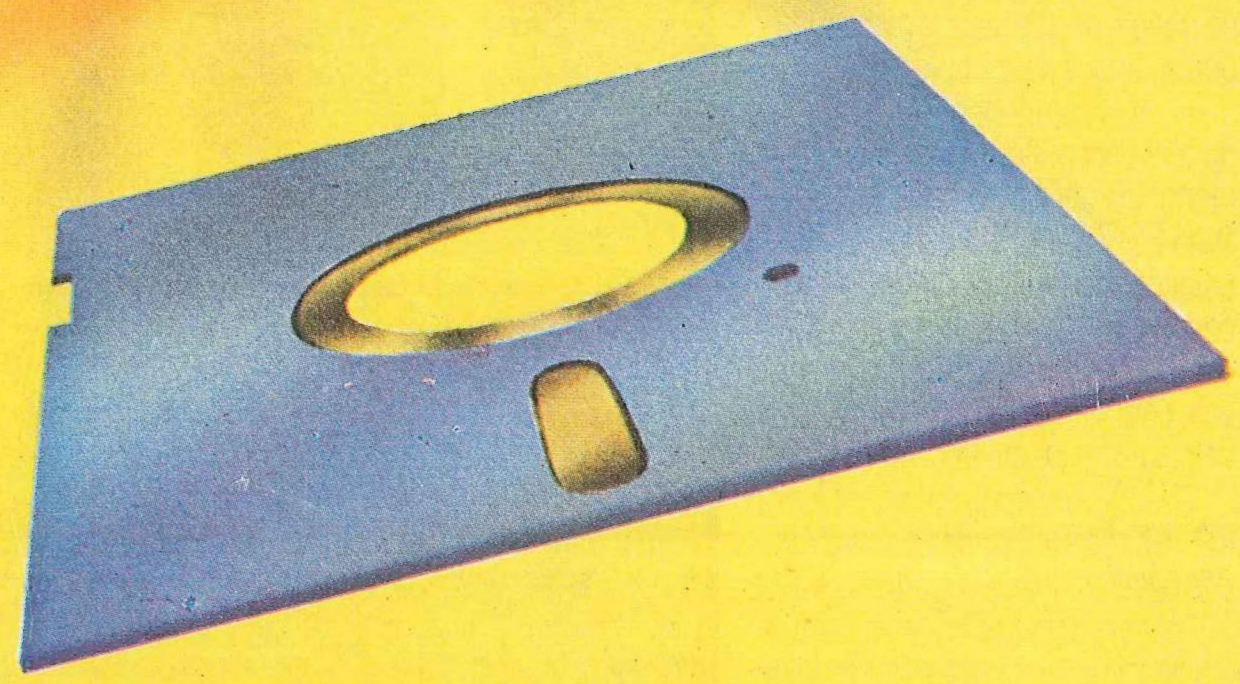
ANO II
NÚMERO 21
Cr$ 4.500,00 (Especial)

Hardware • Software • Computadores • Mini • Micro • Teleprocessamento

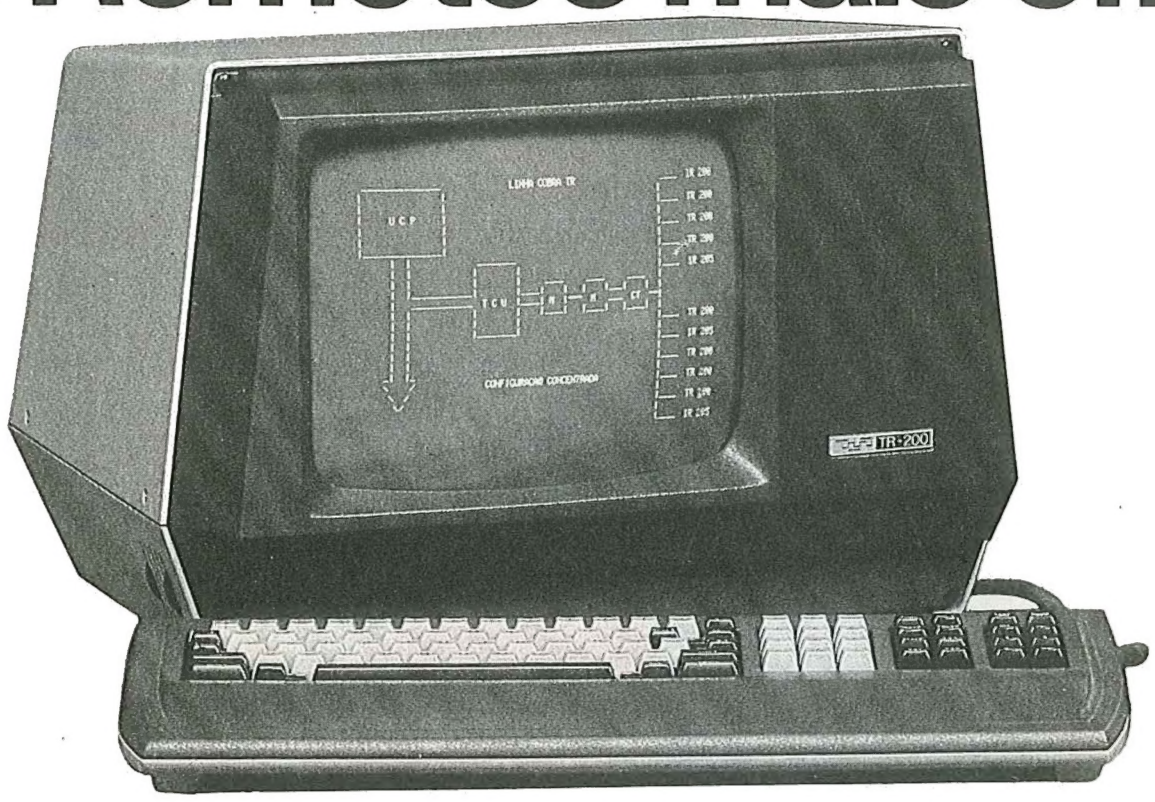
Alta Floresta, Altamira, Boa Vista, Ji-Paraná, Macapá, Manaus, Porto Velho, Rio Branco, Santarém, Sinop (via aérea): Cr$ 5.900,00

EDIÇÃO ESPECIAL

O MELHOR EM SOFTWARE
O MELHOR EM HARDWARE



A Cobra sempre teve os Terminais Remotos mais eficientes.



Agora a Cobra tem os mais bonitos também.

Quando se trata de terminais remotos, eficiência é fundamental, se bem que um pouco de beleza não faz mal nenhum.

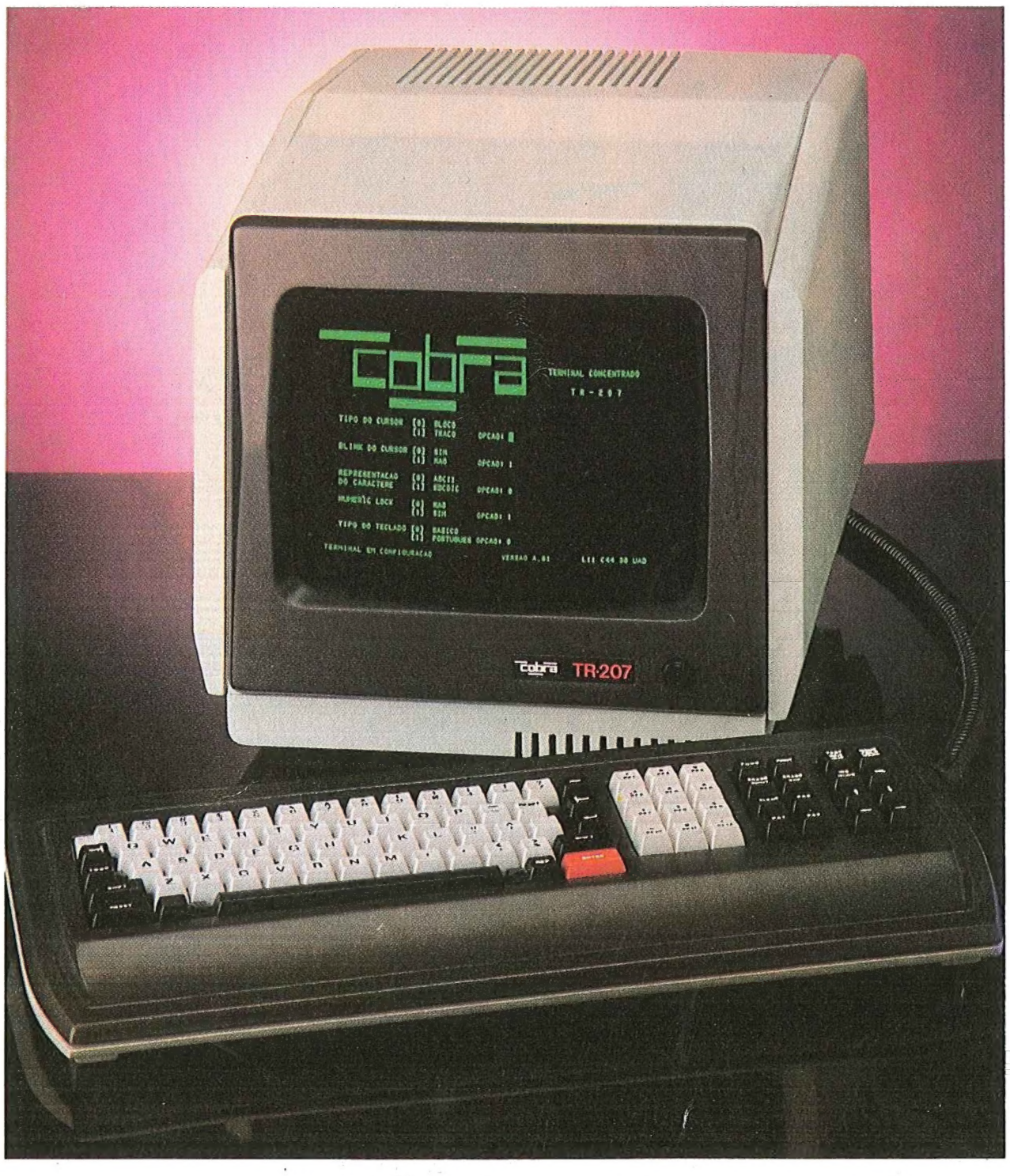
A nova linha de terminais remotos Cobra TR 207 chega para responder às necessidades dos usuários de aplicações on-line com mais eficiência, menor preço e design mais bonito.

Compatível com o sistema de terminais 3270, o TR 207 oferece todas as facilidades disponíveis em sistemas desse tipo.

Os terminais de vídeo podem realizar operações de entrada de dados, programação e consulta, qualquer que seja a aplicação.

Os componentes da linha TR podem ser configurados de diversas maneiras, utilizando protocolo de linha BSC3, comunicando-se com estações de controle de transmissão IBM 3705 ou similar, através de linhas de comunicação em 2 ou 4 fios, com transmissão half-duplex e velocidades de até 9600 bps.

Os terminais de vídeo Cobra TR 207 são suportados pelos métodos de acesso que suportam os IBM 3270, bem como os vários sistemas operacionais e programas-produto.



Qualquer programa aplicativo previamente escrito para os terminais 3270 suporta os terminais de vídeo Cobra TR 207, sem necessidade de alterações.

O TR 207 faz parte da nova linha Cobra 200, que inclui o microcomputador Cobra 210 e o TI 200, terminal inteligente assíncrono.

O microcomputador e os terminais utilizam a mesma mecânica, a mesma fonte de alimentação e o mesmo vídeo, diferindo apenas em algumas características do teclado e da parte eletrônica.

Isso possibilita a transformação de um terminal em micro, bem como permite que o micro funcione como terminal.

Como você vê, além de mais bonita, a nova linha Cobra 200 é também mais inteligente e econômica, porque foi toda pensada para preservar os investimentos do usuário.

Que tal procurar a filial Cobra mais próxima e conhecer a nova linha TR de perto?

CobraTR

Cartão de Consulta nº 145

Chegou o M 301. O micro da Medidata que vale por três.

A Medidata apresenta três ótimas razões que vão levar você a escolher o seu mais novo lançamento: o micro profissional M 301.

1. IBM-PC o M 301 é compatível com o IBM-PC, consagrado como o padrão mundial de micros de 16 bits, colocando à sua disposição a mais ampla e atualizada biblioteca de programas.

2. CP/M-80 o M 301 pode também operar como um micro de 8 bits com CP/M-80, o que representa um aproveitamento muito maior de aplicações e evita o desperdício de investimentos já realizados.

3. MUMPS com o M 301 você tem uma vantagem a mais: o uso do sistema MUMPS, ideal para o desenvolvimento de aplicações administrativas, pela sua simplicidade de programação e eficiência na manipulação de Bancos de Dados, superando com vantagens linguagens tradicionais como Basic e Cobol.

Venha para o M 301 da Medidata, multiplique por três o desempenho de sua empresa e esqueça de vez os problemas de incompatibilidade de linguagens e equipamentos.

medidata
a lógica sob medida

R. J.: Rua Rodrigo de Brito, 13 - Tels.: (021) 542-3737 - 286-5552
S. P.: Av. Paulista, 777 - 1.º andar - Tel.: (011) 288-3522
B. H.: Rua Guajajaras, 40 - s/607 - Tels.: (031) 226-5719 - 226-5045

Oliveira, Murgel

CARACTERÍSTICAS TÉCNICAS

HARDWARE

O hardware do M301 segue padrões atuais de ergonomia, sendo constituído de 3 componentes distintos, Módulo do Sistema, Monitor e Teclado, que podem ser dispostos sobre uma mesa da forma que seja mais conveniente para sua utilização.

Módulo do Sistema

Constitui-se de um gabinete onde são montados: a placa principal, as placas de expansão, os acionadores de disquete ou disco rígido e a fonte de alimentação.

- Microprocessador INTEL 8088 (5MHz)
- Processador aritmético 8087 (opcional)
- 128 ou 256 Kb de memória RAM
- Controle de vídeo cromático
- Interface serial, padrão RS-232C
- Controle de Disquetes
- 5 "slots" para placas de expansão como por exemplo:
  - Controle de disco rígido tipo "Winchester"
  - Placa Z80B
  - Expansão da memória além dos 256 Kb

Placa Z80B (opcional)

Contém um microprocessador Z80B, de 8 bits, e 64 Kb de memória RAM que pode também ser acessada pelo 8088. Através desta placa e do EMULADOR CP/M-80, o M301 pode operar como um micro de 8 bits com CP/M.

SOFTWARE

Além das opções de software de alta qualidade disponíveis através de diversos fornecedores (planilhas eletrônicas, processadores de texto e gráficos, pacotes integrados, etc) a MEDIDATA oferece:

- Sistema Operacional compatível com o MS-DOS da Microsoft
- MUMPS, compatível com o Supermicro M1001 e com os minicomputadores M2001 e M3001
- Basic
- Emulador CP/M-80
- Emulador do terminal de vídeo Medidata
- Utilitários para transferência de arquivos MICRO/MINI/Medidata

Monitor

Monocromático, 12", fósforo verde, tela anti-reflexiva, alta resolução gráfica, sinais RGBI convertidos em 16 níveis de intensidade.

Teclado

Destacável, conectado por cabo espiralado, apoio para as mãos, 30 teclas dispostas num lay-out compatível com o IBM-PC.

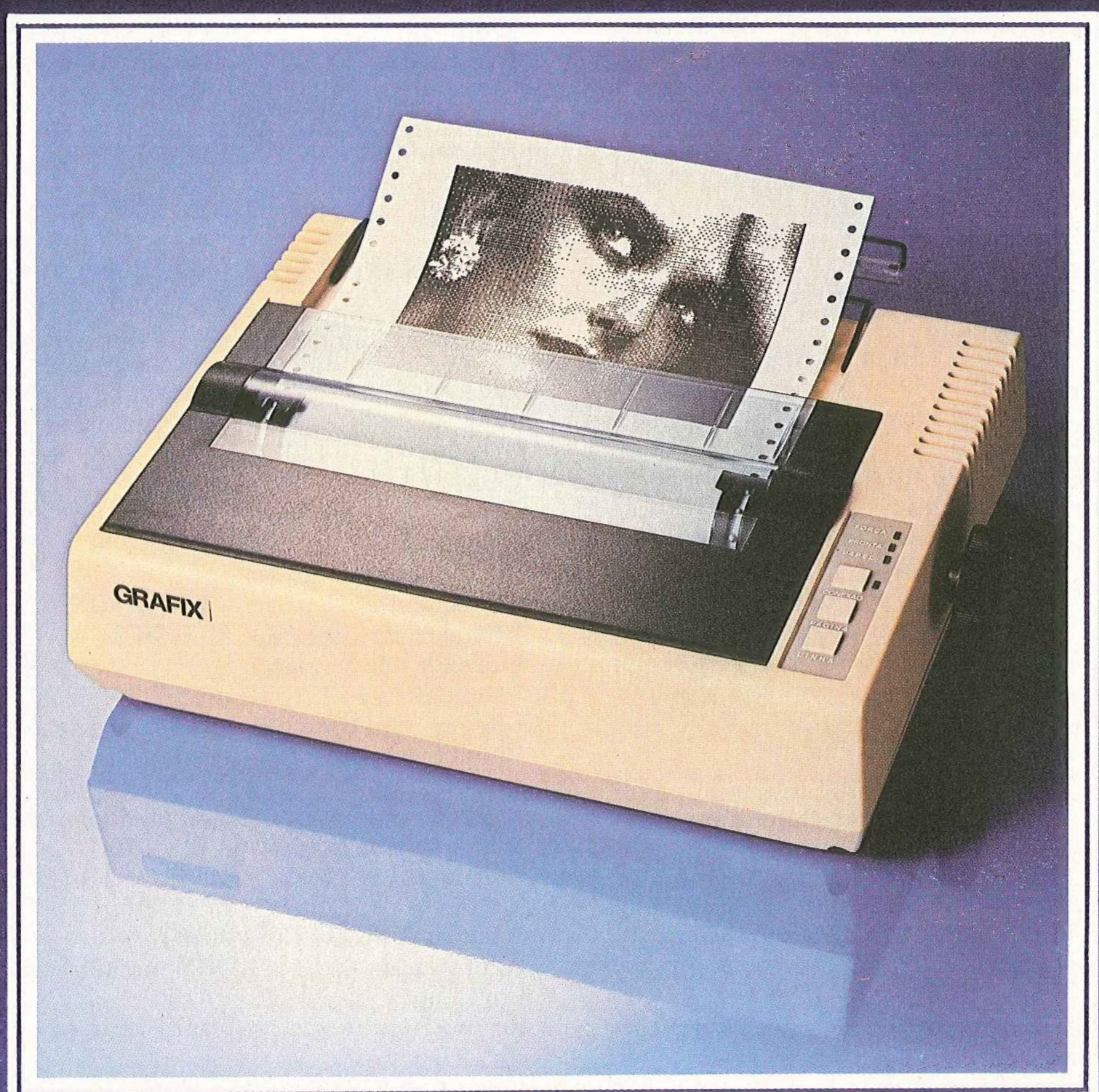
Discos

Acionadores de disquete de 5 1/4", dupla face e dupla densidade, 320 Kb. Acionadores de discos rígidos tipo Winchester de 5 1/4", de 5 ou 10 Mb.

Impressoras

Impressoras matriciais de 100 ou 250 cps, 80 ou 132 colunas, capacidade gráfica e qualidade carta.

Grafix 80
série MX
Impressora Matricial



GRÁFICA

scritta

Especificações

Método de impressão	Impacto por matriz de pontos
Velocidade de impressão	100 CPS
Tempo de avanço de linha	200 ms (1/6 polegada por linha)
Direção de impressão	Bidirecional com busca lógica. Pode ser modificada para unidirecional da esquerda para a direita por comando de programa.
Número de agulhas da cabeça	9
Matriz	9x9
Vida útil da cabeça de impressão	100.000.000 de caracteres
Espaçamento entre linhas	1/6 polegada (normal). Programável em incrementos de 1/72 e 1/216 de polegadas
MTBF da impressora	5.000.000 linhas (excluindo cabeça de impressão)

CARACTERÍSTICAS DE IMPRESSÃO

Conjunto de caracteres	255 caracteres ASCII em matriz 9x9
Tamanho dos caracteres	2,1mm x 3,1mm (normal)
Modos de impressão	Normal Duplo (avança o papel 1/216 e repete a linha) Enfatizado (desloca à direita e duplica a impressão) Duplo enfatizado (combinação das duas anteriores)
Modo gráfico	"Bit-image", em densidade normal horizontal 60 pontos por polegada e, em dupla densidade 120 pontos por polegada

COLUNAS

Normal	80 colunas (10 caracteres por polegada)
Dupla largura	40 colunas (5 caracteres por polegada)
Comprimida	132 colunas (16,5 caracteres por polegada)
Dupla largura comprimida	66 colunas (8,25 caracteres por polegada)

CONTROLE DE FORMULÁRIO

Avanço de linha	Programável de 1 a 85/72
Avanço de formulário	Programável até 127 linhas
Tabulação horizontal	Até 112 posições
Tabulação vertical	Até 64 posições
Salto no picote	Programável no comprimento e intervalos

FORMULÁRIO

Alimentação	Tracionador de formulário contínuo com largura ajustável
Tipo de papel	Formulário contínuo
Largura do papel	101,6mm a 254mm (4 a 10 polegadas)
Espessura do papel	0,3mm (máximo)
Número de cópias	Um original mais duas cópias

INTERFACE

Tipo	Interface paralela tipo Centronics (opcionais RS 232, IEEE 488)
Velocidade de transferência de dados	4.000 bytes por segundo
Sincronização	Pulsos STROBE externos
Protocolo	Por ACK ou BUSY
Nível lógico	Compatível com TTL

FITA DE IMPRESSÃO

Apresentação	Cassete
Cor	Preta
Vida útil	3.000.000 de caracteres

CHAVES/INDICADORES/DETECTORES

Indicadores	Força, pronta, papel e conexão
Chaves	Liga/desliga, conexão, avanço de página e avanço de linha
Detectores	Campainha interna acusa falta de papel e condições de erro

CONDIÇÕES AMBIENTAIS

Temperatura de operação	$5^{o}C$ a $35^{o}C$
Temperatura de armazenamento	$-30^{o}C$ a $70^{o}C$
Umidade de operação	5% a 90%
Umidade de armazenamento	0% a 95%
Choque em operação	1 G (menos de 1 ms)
Choque em armazenamento	2 G (menos de 1 ms)
Vibração em operação	0,25 G, 55 Hz (máximo)
Vibração em armazenamento	0,5 G, 55 Hz (máximo)

ALIMENTAÇÃO/CONSUMO

Alimentação	110V ou 220V, 49,5 a 60,5 Hz
Consumo	110 VA
Corrente	Menor 1A

CARACTERÍSTICAS FÍSICAS

Altura	107mm
Largura	374mm
Profundidade	305mm
Peso	5,5 Kg

AUTO TESTE

Pressione a tecla LINHA enquanto liga a impressora. O auto teste imprime todos os caracteres que estão na memória ROM.

Especificações sujeitas a modificações sem prévio aviso.

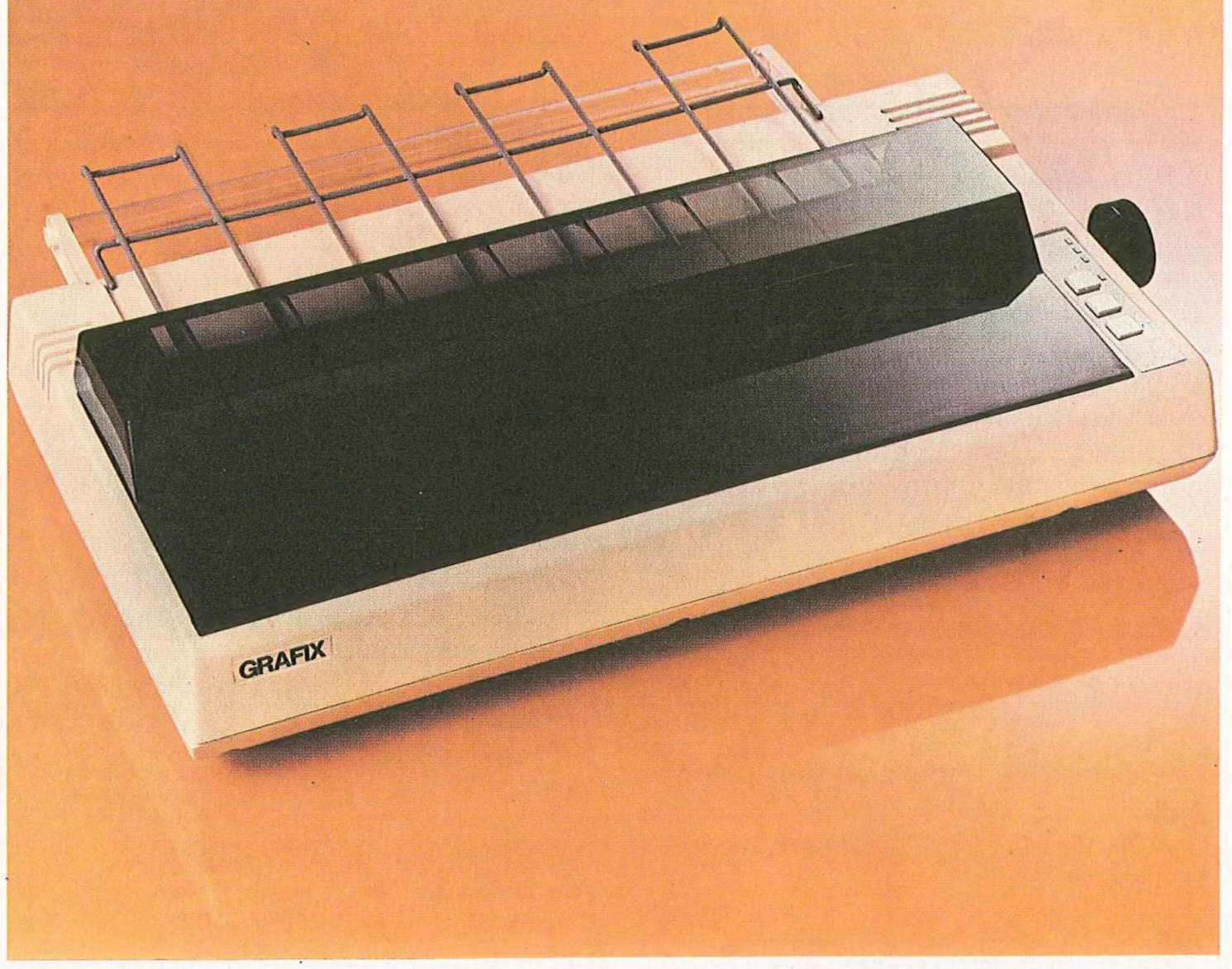
GRAFIX

SCRITTA ELETRÔNICA LTDA.

RUA HELIÓPOLIS N.° 61/67 - VILA LEOPOLDINA - SÃO PAULO - CEP 05318
FONES: 831-9912 - 831-5177 - TELEX: (011) 25056 SCTT BR

PSP

Grafix 100
série MX
Impressora Matricial



GRÁFICA

scritta

Especificações

Método de impressão	Impacto por matriz de pontos
Velocidade de impressão	100 CPS
Tempo de avanço de linha	200 ms (1/6 polegada por linha)
Direção de impressão	Bidirecional com busca lógica. Pode ser modificada para unidirecional da esquerda para a direita por comando de programa.
Número de agulhas da cabeça	9
Matriz	9x9
Vida útil da cabeça de impressão	100.000.000 de caracteres
Espaçamento entre linhas	1/8", 1 1/6". Programável em incrementos de 1/72 e 1/216 de polegadas
MTBF da impressora	5.000.000 linhas (excluindo cabeça de impressão)

CARACTERÍSTICAS DE IMPRESSÃO

Conjunto de caracteres	255 caracteres ASCII em matriz 9x9
Tamanho dos caracteres	2,1 x 3,1mm
Modos de impressão	Normal Dupla (avança o papel 1/216 e repete a linha) Enfatizado (desloca à direita e duplica a impressão) Dupla enfatizado (combinação das duas anteriores)
Modo gráfico	"Bit-image", em densidade normal horizontal 60 pontos por polegada e, em dupla densidade 120 pontos por polegada

COLUNAS

Normal	136 colunas (10 caracteres por polegada)
Normal expandido	68 colunas (5 caracteres por polegada)
Comprimido	233 colunas (16,5 caracteres por polegada)
Comprimido expandido	116 colunas (8,25 caracteres por polegada)

TABULAÇÃO DE FORMULÁRIO

Avanço de linha	Programável de 1 a 85/72
Avanço de formulário	Programável até 127 linhas
Tabulação horizontal	Até 12 posições
Tabulação vertical	Até 8 posições
Salto no picote	Programável no comprimento e intervalos

FORMULÁRIO

Alimentação	Tracionador de formulário contínuo com largura ajustável
Tipo de papel	Formulário contínuo
Largura do papel	101,6 a 393,7mm (4 a 15,5 polegadas)
Espessura do papel	0,3mm (máximo)
Número de cópias	Um original mais duas cópias

INTERFACE

Tipo	Interface paralela tipo Centronics (opcionais RS 232 e IEEE 488)
Velocidade de transferência de dados	4.000 bytes por segundo
Sincronização	Pulsos STROBE externos
Protocolo	Por ACK ou BUSY
Nível lógico	Compatível com TTL

FITA DE IMPRESSÃO

Apresentação	Cassete
Cor	Preta
Vida útil	3.000.000 de caracteres

CHAVES/INDICADORES/DETECTORES

Indicadores	Força, pronta, papel e conexão
Chaves	Liga/desliga, conexão, avanço de página e avanço de linha
Detectores	Campainha interna acusa falta de papel e condições de erro

CONDIÇÕES AMBIENTAIS

Temperatura de operação	5°C a 35°C
Temperatura de armazenamento	- 30°C a 70°C
Umidade de operação	5% a 90%
Umidade de armazenamento	0% a 90%
Choque em operação	1 G (menos de 1 ms)
Choque em armazenamento	2 G (menos de 1 ms)
Vibração em operação	0,25 G, 55 Hz (máximo)
Vibração em armazenamento	0,5 G, 55 Hz (máximo)

ALIMENTAÇÃO/CONSUMO

Alimentação	110V ou 220V, 49,5 a 60,5 Hz
Consumo	100 VA
Corrente	Menor 1 A

CARACTERÍSTICAS FÍSICAS

Altura	133mm
Largura	592mm
Profundidade	393mm
Peso	10 Kg

AUTO TESTE

Pressione a tecla LINHA enquanto liga a impressora. O auto teste imprime todos os caracteres que estão na memória ROM.

Especificações sujeitas a modificações sem prévio aviso.

GRAFIX

SCRITTA ELETRÔNICA LTDA.

RUA HELIÓPOLIS N.° 61/67 - VILA LEOPOLDINA - SÃO PAULO - CEP 05318
FONES: 831-9912 - 831-5177 - TELEX: (011) 25056 SCTT BR

PSP

Interface

EDIÇÃO ESPECIAL

Editor Científico
Cesar da Costa

Diagramação e arte-final
Ruy Pereira/Sandra M. de Oliveira

Publicidade
Supervisão geral
Mariangela B. Losasso
Rio de Janeiro
José Augusto Nunes Rodrigues
Tel: 221-4293
São Paulo
Suzana de Moraes Gregores, Carmem Lúcia M.C. e Silva, Lúcia Helena Macedo de Paula, Ana Cristina Koda
Indianópolis
Alameda dos Anapurus, 859 - tel. (011) 241-8629 e (011) 542-9949

Tráfego
Simone Regina Paes - SP

Assinaturas
São Paulo: José Roberto Correa
Rio de Janeiro: Aloisio C. da Rocha
Outras capitais – Ildo Idolar Peres

Preços:
Brasil - Cr$ 25.000,00 (1 ano) e Cr$ 50.000,00 (2 anos).
Exterior - US$ 50 (1 ano) e US$ 100 (2 anos)

Distribuição
Brasil: DINAP - Distribuidora Nacional de Publicações; Estrada Velha de Osasco, 132, tel. (011) 268-2522, telex: 33670 - ABSA.
Portugal: Distribuidora Jardim de Publicações Ltda., Quinta Pau Varais, Azinhaga dos Fetais, 2685, Camarate, Lisboa.

INTERFACE é uma publicação de propriedade da Editora Interface Ltda. Redação, Adminstração e Publicidade: Alameda dos Anapurus, 859, Indianópolis, 04087 - São Paulo - tel. (011) 241-8629/542-9949

Registro no 21.º Ofício de Notas do Rio de Janeiro sob n.º 810.965.755.

Impressão: CLY - Rua Cadete, 209

Todos os direitos reservados; proíbe-se a reprodução parcial ou total dos textos e ilustrações desta publicação, assim como traduções e adaptações, só podendo ser reproduzidos com fins comerciais, mediante autorização prévia. Os artigos assinados são de responsabilidade do autor, as opiniões emitidas não são necessariamente coincidentes com as da revista.

Dois anos trabalhando pela informação técnica com seriedade, profissionalismo e competência

Há dois anos, na III Feira Internacional de Informática, no Rio de Janeiro, lançávamos a Revista Interface.

". . . Não somos uma revista só de microcomputadores, somos uma revista de informática, com linguagem e vida própria adquiridas, nas salas de aula, ministrando cursos de especialização, treinamento de pessoal, laboratórios de pesquisa e de manutenção, em CPD, na indústria e no nosso dia-a-dia (notas do editor Interface nº1).

A partir dali, tínhamos definido nossa linha editorial, que mantivemos até hoje, numa constante atualização. A microinformática estava destinada a ocupar um lugar de destaque entre as ciências da computação, figurando inclusive como uma das mais importantes. Naquela época, alguns profissionais, estudantes e usuários, provavelmente só tinham ouvido falar em microcomputadores e alguma coisa sobre os chips de silício, que tornaram a microinformática possível. O que ainda não haviam percebido era que dentro dos próximos anos os técnicos e usuários profissionais, para dominarem os microcomputadores (programação e projeto), deveriam primeiro aprender como eles trabalhavam internamente (software básico e hardware), a fim de que, pudessem tirar maior vantagem deles, desenvolvendo novos programas ou otimizando os existentes.

Se as pessoas tivessem que entender como os automóveis funcionam para poder dirigi-los, o número de automóveis nas estradas seria automaticamente reduzido. Contudo, enquanto os automóveis se beneficiaram com aproximadamente cem anos de refinamento, proporcionado pela engenharia, no Brasil os microcomputadores eram relativamente novos. Exatamente como os primeiros automóveis, tinham suas peculiaridades quanto ao funcionamento, tais como a manivela, e eram muito menos acessíveis para a média das pessoas do que se tentava divulgar.

Havia um grande número de razões para muitos técnicos e usuários aprenderem a programar seus próprios micros, apesar da disponibilidade de software existente no mercado. Uma das razões estava no fato de individualizar o microcomputador para que ele trabalhasse exatamente da maneira que o programador queria que ele trabalhasse, fosse em aplicações comerciais, domésticas ou de engenharia. Outra razão seria o desenvolvimento de produtos que impulsionariam a independência da tecnologia nacional de software e hardware.

Hoje, dois anos depois, pouca coisa mudou no cenário nacional. A venda de microcomputadores no varejo como automóveis não obteve grandes resultados. Porém, felizmente os microcomputadores nacionais começaram a se beneficiar com o refinamento proporcionado pela microinformática, técnicos e usuários profissionais preocupam-se mais em conhecê-la. É o amadurecimento da microinformática ao nível de usuário final.

A mesma certeza de dois anos atrás, que não estávamos lançando mais uma revista de informática, nos incentivou a publicar a nossa primeira edição especial: **"O MELHOR DE INTERFACE EM HARDWARE E SOFTWARE"**.

Esta edição especial é a primeira de uma série que publicaremos e será transformada em livros técnicos por uma grande editora científica, possibilitando que, as escolas técnicas, Universidades e cursos livres, adotem Interface como livro didático, auxiliando na formação e especialização dos profissionais brasileiros.

Nossa iniciativa é pioneira no mercado nacional, similar ao que a revista BYTE americana proporcionou ao mercado americano divulgando e popularizando a microinformática. Não foi tarefa fácil escolher o melhor em hardware e software editado nesses dois anos. Assim sendo, esperamos que esta iniciativa seja tão bem sucedida quanto foi o lançamento de Interface.

Cesar da Costa

Interface - cad.

Ponha o seu micro para trabalhar

você terá mais tempo para descansar...

Assinale as publicações que deseja receber:

- ☐ **Alexander.** Programação em Assembler e Linguagem de Máquina. **17.200,00**
- ☐ **Barden.** Microcomputadores para Aplicações Comerciais. **19.600,00**
- ☐ **Chance.** 30 Programas em Basic para Computadores Pessoais. **21.300,00**
- ☐ **Chantler.** Técnicas e Prática de Programação. **13.400,00**
- ☐ **Helms.** Guia de Linguagens de Computadores. **10.800,00**
- ☐ **Hergert.** Basic para Aplicações Comerciais. **14.500,00**
- ☐ **Kember.** Aplicações do Computador na Medicina. **17.600,00**
- ☐ **Knight.** Implantação de Micros e Minicomputadores Comerciais. **11.400,00**
- ☐ **Kuecken.** Aplicações de Microprocessadores. **22.900,00**
- ☐ **Lucas.** Como Lidar com o Computador. **13.000,00**
- ☐ **Moreira.** Criança Também faz Programas 2. ed. **11.400,00**
- ☐ **Pereira.** BASIC Básico 4. ed. **13.600,00**
- ☐ **Pereira.** BASIC para Micros Pessoais 2. ed. **12.300,00**
- ☐ **Sawusch.** 1001 Aplicações para o Seu Computador Pessoal. **20.000,00**
- ☐ **MICROBITS.** Uma Publicação Bimestral com informações e programas para Microcomputadores TK-82-83-85, CP-200 e compatíveis

Assinatura: Cr$ 13.800,00 **Nº Avulso: Cr$ 3.100,00**

Assinale aqui a sua opção de compra:

- ☐ Cheque Nominal à Editora Campus Ltda.
  Nº . . . . . Banco . . . . . .
  (o porte é por nossa conta e sua encomenda chega mais rápido)
- ☐ Reembolso Postal

Envie o seu cupom ainda hoje para:

Editora Campus Ltda.

Caixa Postal 3954 – CEP 20001 –
Rio de Janeiro – RJ

Nome . . . . . .
Endereço . . . . . .
Cidade . . . . . .
CEP. . . . . Estado. . . . . .

Importante: Preços válidos até **31/12.84.**

EDITORA CAMPUS

Otavio Studart

O MELHOR EM SOFTWARE

1 - MICROINFORMÁTICA: O QUE É? ONDE SE ENCONTRA?

Tarcísio Neves da Cunha

(Revistas Interface Vol. 1, n.º 1, págs. 26 a 32; Vol. 1, n.º 2, págs. 32 a 36)

1.1 – Conceituação básica
1.2 – Propriedades
1.3 – Quanto a capacidade de memória principal
1.4 – Quanto ao passo de transferência
1.5 – Quanto ao número de bits dos operadores da ALU
1.6 – Quanto a versatilidade de processamento
1.7 – Quanto a capacidade de entrada/saída
1.8 – Unidades de apoio
1.9 – Memórias
1.10 – Interfaces e controladores
1.11 – Coprocessador
1.12 – Software encapsulado
1.13 – Aplicações
1.14 – Processamento de dados
1.15 – Desenvolvimentais
1.16 – Processamento de sinais
1.17 – Controle de processos

2 - O SISTEMA OPERACIONAL CP/M

Valdelírio P. S. Filho

(Revistas Interface Vol. 1, n.º 3, págs. 38 a 40; Vol. 1, n.º 4, págs. 30 a 33; Vol. 1, n.º 5, págs. 32 a 35; Vol. 1, n.º 6, págs. 23 a 25; Vol. 1, n.º 8, págs. 28 a 30)

Cláudio Esperança

(Revistas Interface Vol. 2, n.º 16, págs. 31 a 33; Vol. 2, n.º 17, págs. 31 a 33; Vol. 2, n.º 18, págs. 29 a 31)

2.1 – Histórico
2.2 – A estrutura
2.3 – Referência de arquivos
2.4 – Posicionamento em disco e processo de carga
2.5 – O BIOS
2.6 – O IOBYTE
2.7 – As rotinas do Bios
2.8 – O Diretório
2.9 – O FCB
2.10 – As tabelas de parâmetro do disco
2.11 – O DPH
2.12 – O BDOS
2.13 – Chamada das funções
2.14 – As funções
2.14-01 – CP/M 3.0
2.14-02 – O LOGIN VECTOR
2.14-03 – Reinicializa o sistema de disco
2.14-04 – Seleciona o disco
2.14-05 – Abre arquivo
2.14-06 – Fecha arquivo
2.14-07 – Busca a primeira
2.14-08 – Busca a próxima
2.14-09 – Apagar arquivo
2.14-10 – Lê seqüencialmente
2.14-11 – Escreve seqüencialmente
2.14-12 – Criar arquivo
2.14-13 – Rebatizar arquivo
2.14-14 – Retorna o LOGIN VECTOR
2.14-15 – Retorna o disco corrente
2.14-16 – Estabelece o endereço da área de DMA
2.14-17 – Protege disco contra escrita
2.14-18 – Retorna o READ ONLY
2.14-19 – Estabelece atributos de arquivos
2.14-20 – Retorna o DISCK PARAMETER BLOCK (DPB)
2.14-21 – Lê ou modifica o número do usuário
2.14-22 – Lê randomicamente
2.14-23 – Grava randomicamente
2.14-24 – Retorna o tamanho do arquivo
2.14-25 – Estabelece o número de registro randômico
2.14-26 – Reinicializa o disco lógico
2.14-27 – Grava randomicamente registros com zeros
2.15 – Utilizando as funções BDOS
2.15-01 – Entendendo o programa

MICRO INFORMÁTICA: O QUE É? ONDE ENCONTRÁ-LA?

"É mister tomar para nós um mercado que é nosso, sob pena de nos encontrarmos, a médio prazo, como meros operadores de pacotes fechados, adaptando a eles as nossas condições e não o contrário, que é a saudável adaptação da tecnologia às circunstâncias de cada caso".

Mesmo em tempos de paz, a grande força propulsora da Eletrônica é a guerra. A década de 40 marca o início de um desenvolvimento explosivo desta disciplina nos países ricos, como resultado de esforços titânicos para menter sua influência sobre as outras nações, a fim de garantir crescentes dividendos comerciais via colonialismo econômico. Fecha-se assim um negro círculo vicioso: riqueza gerando poder e tecnologia, para gerarem mais riqueza.

Evidentemente o abismo técnico-científico verificado entre as nações ricas e as restantes é cada vez maior, dificultando mas não impossibilitando a equiparação. Com criatividade e determinação, objetivos bem definidos podem ser atingidos e assimilados em espaço de tempo viável, servindo como ponta de lança para outros desbravamentos. Foi assim que o Brasil fez com seu território e sua independência política, e é assim que fará com sua independência tecnológica. O delicado aspecto financeiro da questão é relevante, mas penso que não é determinante, dada a abundância dos recursos internos e a crescente capacitação da nossa sociedade para administrá-los. Além do mais, se os grandes impérios se fizeram sem financiamento externo, quanto mais o Brasil, cujas ambições estão longe de ser hegemônicas.

Dentro deste contexto a Informática tem seu papel a cumprir, pulverizado na consciência de cada profissional da área. É mister tomar para nós um mercado que é nosso, sob pena de nos encontrarmos, a médio prazo, como meros operadores de pacotes tecnológicos fechados, adaptando a eles as nossas condições e não o contrário, que é a saudável adaptação da tecnologia às circunstâncias de cada caso.

É assim que chegamos ao núcleo da questão abordada neste trabalho: a Informática encontrou na Eletrônica

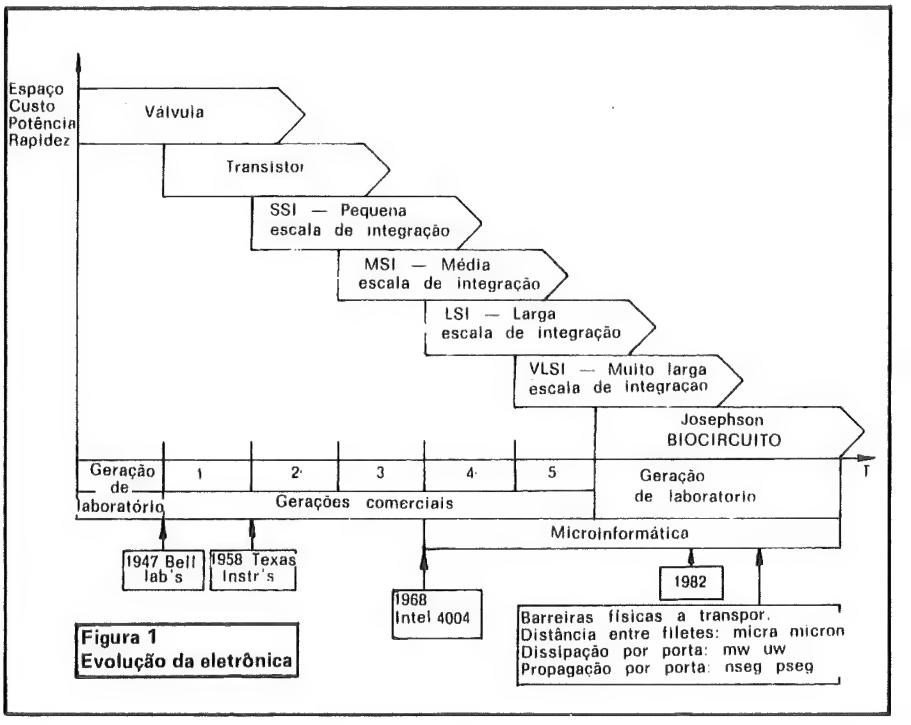


Figura 1
Evolução da eletrônica

uma base sólida, onde apoiou seu próprio desenvolvimento. Com o aperfeiçoamento da técnica dos semicondutores, os circuitos eletrônicos "implodiram" em tamanho surgindo a microeletrônica, pequena também em custo, fomentando na indústria a produção de componentes com capacidade de processamento. Daí o nascimento do que chamamos **Microinformática**, que embora recente já penetrou nos diversos ramos da cultura moderna.

CONCEITUAÇÃO BÁSICA

Em que pese a equivalência de conceitos da Microinformática em relação à informática, emitiremos alguns que nos parecem mais pertinentes, a fim de padronizar a terminologia, de resto nem sempre clara na literatura corrente. Não nos preocuparemos outrossim, com definições formais, preferindo conceitos precisos.

Partiremos do geral para o específico.

- **Sistema** é todo artefato capaz de exercer transformações sobre uma ou mais entidades. Observe que esta é uma generalização do conceito físico de máquina: é só substituir "entidades" por "formas de energia". Por isso é lícito chamar um processador de "máquina". Senão vejamos:
- **Sistema Controlável** é o sistema dotado de regras paramétricas de transformação, de modo a se poder exercer controle sobre o comportamento do sistema.

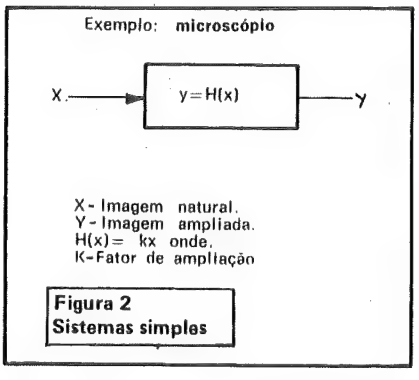
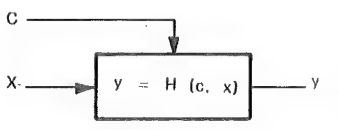


Figura 2
Sistemas simples

- **Sistema Programável** é o sistema dotado de mais de uma regra de transformação, cuja seleção dá-se dinamicamente através, das próprias entidades a ele submetidas. Assim, as entidades nos sistemas programáveis se classificam em dois grupos:
- **Instrução** ou **Operador** é a entidade de um sistema programável especifica para selecionar uma dentre todas as regras de transformação disponíveis.
- **Dado** ou **Operando** é a entidade de um sistema programável efetivamente passível de sofrer transformação.



Exemplo: **aparelho de TV**

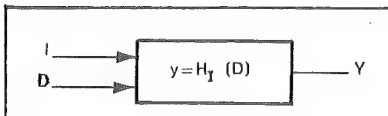
X - Ondas eletromagnéticas captadas pela antena.
C - Controle de:
Entrada — Seletor de canais, que isola apenas uma dentre as várias freqüências captadas.
Saída de som — volume, tom, etc.
Saída de imagem — intensidade, brilho, sintonia H-V, etc.
Y - Som e imagem.
H(c,x) Função complexa, resultado do encadeamento de diversos estágios: sintonia amplificação, varredura, etc.
Observe-se que X não fornece indicações sobre volume ou intensidade da imagem. Tais informações são **parâmetros** próprios das funções de transferência implementadas.

Figura 3
Sistema controlável

E mais: quando sistemas programáveis são concebidos para tratamento de "INFORMAÇÕES". segue-se que:

- **Informação** é a designação genérica para DADO e INSTRUÇÃO, e
- **Informática** é toda a atividade voltada para o aperfeiçoamento dos sistemas programáveis aplicados à INFORMAÇÃO.

Tais atividades estão estratificadas, segundo os conceitos até agora emitidos, em três grupos: o "ATUADOR",



Onde:
D — Dados de entrada } Informações
Y — Dados de saída }
Cada H_I é uma Função de Transferência única que opera sobre D.
Dentre os vários H disponíveis, apenas um é selecionado de cada vez, pela entrada I.

Figura 4
Sistema programável

que executa as transformações; as entidades do tipo "INSTRUÇÕES" e as do tipo "DADOS", cobrindo amplo espectro de especialidades profissionnais. Engenheiros eletrônicos e de sistemas constroem os atuadores, segundo uma gama de funções de transferência estabelecidas por projetistas e arquitetos de sistemas. A entidade **instruções** cobre desde linguagem de máquina até aplicativos, empregando todo pessoal da área do software. A entidade **dados** congrega os especialistas de cada ramo de aplicação na confecção de metodologias, que o pessoal de software e engenharia de aplicação converte em algoritmo.

- **Microinformática** é a informática que usa ATUADORES altamente integrados, graças à microeletrônica. Observe-se desde já, que a recíproca não é verdadeira. Todos os computadores modernos e demais áreas apoiadas pela eletrônica, usam microcomponentes. Apenas um subconjunto da microeletrônica é dedicado à microinformática.

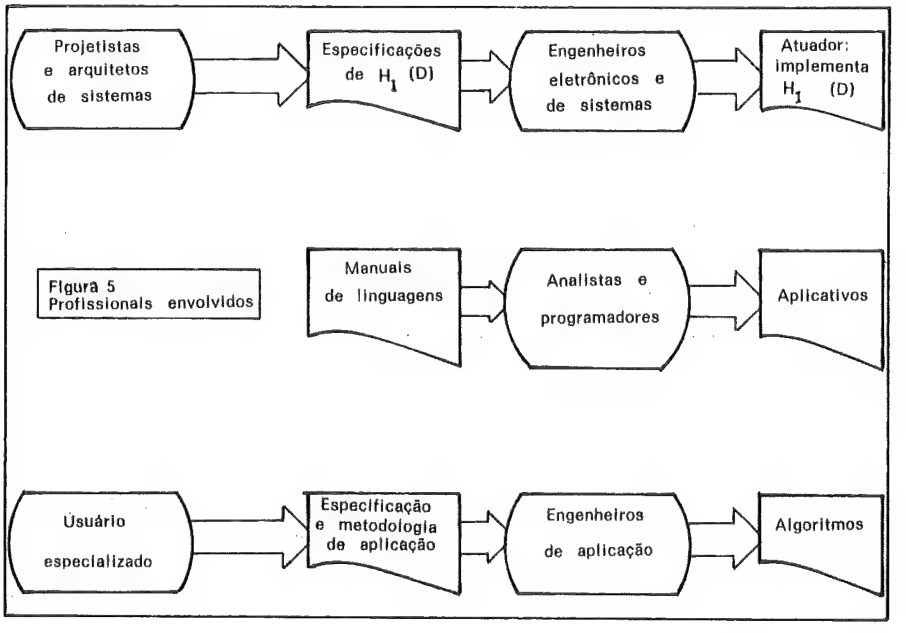


Figura 5
Profissionais envolvidos

• **Computador** é o nome genérico dado aos componentes físicos dos sistemas programáveis de processamento de informações. Podemos então estabelecer a relação terminológica e conceitual entre os componentes do sistema e do computador. A implementação do ATUADOR é denominada PROCESSADOR, com seu banco de registradores, unidade de controle e temporização e unidade lógica e aritmética.

As **ENTIDADES** ou **INFORMAÇÕES** pertencem ao domínio das idéias e só podem ser submetidas ao computador via PERIFÉRICOS de entrada, capazes de dar-lhes uma representação física reconhecível pelo processador. Da mesma forma, após sofrerem transformação, voltam ao mundo exterior através de PERIFÉRICOS de saída, que restabelecem sua representação natural.

As informações em representação interna são armazenadas nos dispositivos de MEMÓRIA:

a) **Primária** ou **Principal**, quando na iminência de serem submetidas ao processador.

b) **Secundária** ou **Auxiliar**, quando da necessidade de arquivamento.

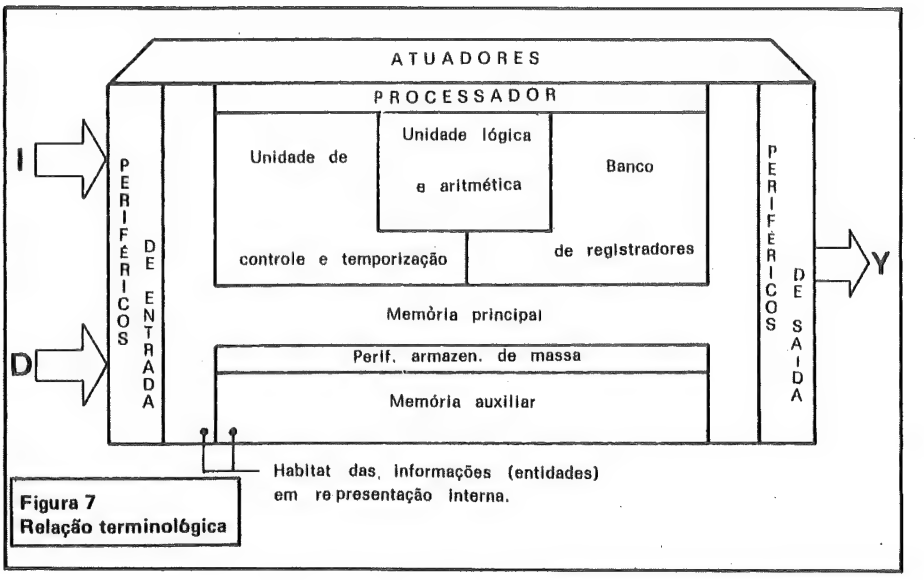


Figura 7
Relação terminológica

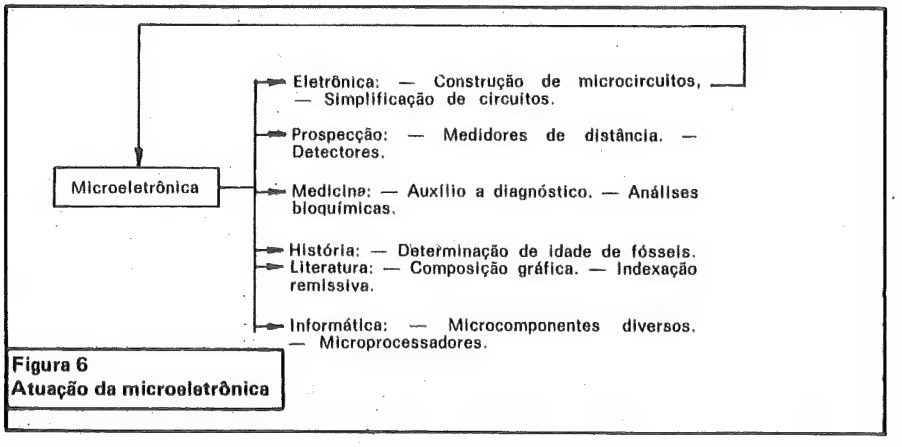


Figura 6
Atuação da microeletrônica

Uma contribuição da microeletrônica está na integração de diversos componentes de um circuito em um único microcomponente. Por exemplo, a unidade de controle e temporização pode ser implementada em uns poucos microcomponentes. Os dispositivos de memória já são todos integrados nas máquinas modernas de qualquer porte.

Alguns fabricantes de microcomponentes colocaram à disposição do mercado diversas implementações de unidades lógicas e aritméticas, atuando geralmente em 2 ou 4 bits, mas com capacidade de encadeamento em série para formar o tamanho de palavra que for desejado. Tais componentes são chamados de BIT-SLICE PROCESSORS, por serem fatias de número fixo de bits, encadeáveis logicamente lado a lado. Esta técnica é muito usada em máquina de porte mini e médio, como o MB 8000 da SISCO e o COBRA 500.

No âmbito deste trabalho, só consideraremos microinformática ao subgrupo de sistemas que usa ATUADOR completamente microminiaturizado. Assim, do ponto de vista de sistema:

Microprocessador é o ATUADOR de sistemas programáveis aplicados a MICROINFORMÁTICA. Logo, do ponto de vista funcional:

Microprocessador é um **componente eletrônico** de larga escala de integração (LSI e VLSI), que implementa as funções de um processador, quais sejam: banco de registradores, unidade lógica e aritmética e unidade de controle e temporização. Do ponto de vista eletrônico:

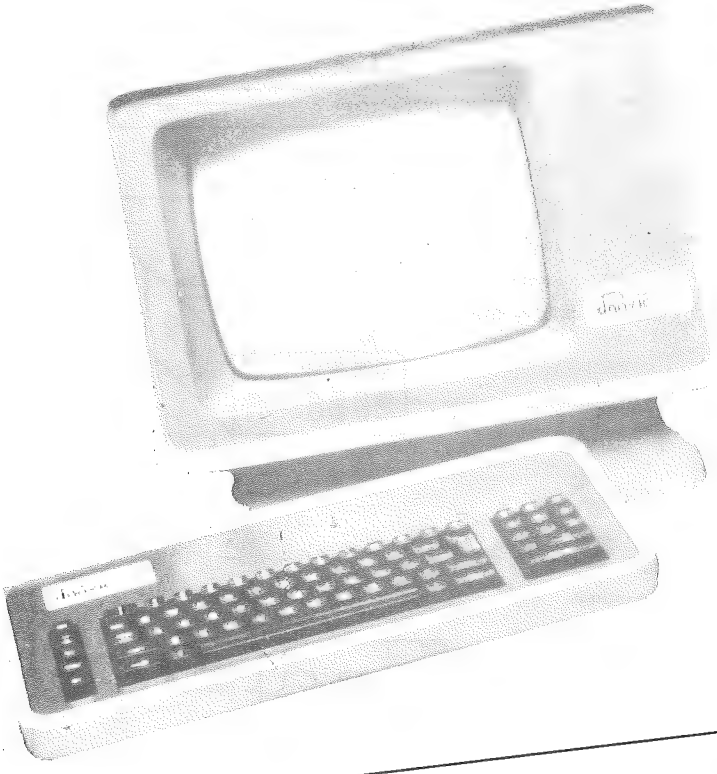
Microprocessador é uma **micromáquina eletrônica**, digital, seqüencial, síncroma. Evidemente, nenhum componente eletrônico funciona sozinho. Nem o microprocessador. Necessita-se de um circuito externo para isso. Além do mais, para se transformar um microprocessador em microcomputador, precisa-se adicionar memória e interfaces para periféricos. Se conseguirmos fazer isto tudo numa única placa de circuito impresso, obteremos o que se convencionou chamar de:

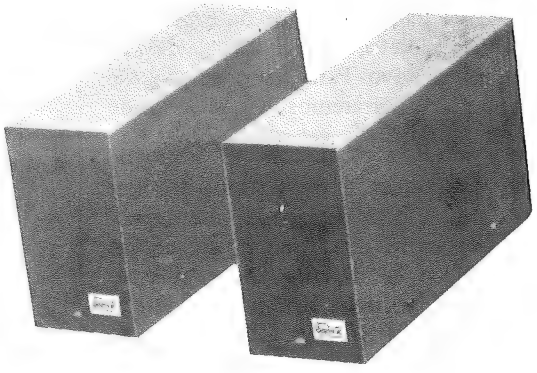
• **Single Board Computer** (SBC) é uma placa de circuito impresso contendo um microcomputador, memória primária e alguma interface de entrada/saída. É só conectar o(s) periférico(s) recomendados, uma fonte de alimentação e temos um microcomputador. Alguns já vêm até com algum software para programação (geralmente MONTADOR HEXADECIMAL DE LINGUAGEM DE MÁQUINA, ou ASSEMBLER, ou INTERPRETADOR BASIC). A limitada capacidade dos SBCs não é relevante, pois apresentam facilidades de expansão em outras placas.

A tecnologia foi ainda mais além. Tendo em vista necessidades de processamento com limites estreitos bem definidos, como por exemplo controle dos sistemas de um automóvel ou de uma linha de produção, conseguiu-se introduzir o chamado **Single Chip Microcomputer** que é um componente eletrônico que integra um processador, alguma memória primária e facilidade de entrada/saída.

Estes componentes são ainda mais limitados que os SBCs devido aos obje-

Quando você compra o Danvic Caçula, leva um microcomputador por inteiro.





Cada vez que V. precisa aumentar os recursos dos microcomputadores que existem por aí, precisa adaptar as famosas placas. Placa para discos, placa para impressora, placa para ampliação de memória, placa para sistema CP/M, placa para vídeo 80x24. Então, descobre que a fonte não está alimentando adequadamente todas as interfaces. E aumenta a fonte. E põe um ventilador para resfriar a fonte que esquenta. No fim, descobre que comprou uma grande dor de cabeça.

O DANVIC CAÇULA foi originalmente projetado com todas as saídas necessárias para executar tudo de acordo com suas necessidades e já é dimensionado, em termos de alimentação, para isso. Vem com dois microprocessadores Z80-A de 64K e clock de 4MHz (4 vezes mais rápido que qualquer similar ao Apple), vídeo de fósforo verde 80x24, capacidade para até 4 floppies para discos de 5"1/4 com dupla densidade e face dupla ou simples (determina-se densidade e face diretamente pelo teclado), saída para impressora, saída para telecomunicação, sistema CP/M 2.2, linguagem BASIC, editor de textos e vários programas utilitários. Além disso, o CAÇULA é totalmente compatível com todos os membros da família DANVIC, como o DV-600 ou o DV-3000 e também com outros Microcomputadores do mercado que utilizam CP/M (todo software feito para os outros poderá trabalhar no CAÇULA com vantagens).

Opcionalmente, pode ser fornecido com linguagem COBOL, FORTRAN ou PASCAL e diversos programas aplicativos específicos. E você ainda conta com todo o apoio técnico de quem conheça a fundo seu microcomputador.

Entre em contato hoje mesmo com a DANVIC e veja, na frente de seus olhos e sem truques, o que o CAÇULA pode fazer por você e sua empresa.

danvic computadores

DANVIC S.A.: Rua Conselheiro Nébias, 1409 — CEP 01203 — Telex 1123888 CICP BR Tel.: (011) 221.6033 — São Paulo

Filias e representantes em todo o Brasil

tivos a que se propõem, atingindo drástica redução de custos. Alguns admitem expansões externas também.

A grande contribuição da microeletrônica para a informática trouxe um não menor benefício para a própria eletrônica. Microrprocessadores reduzem dramaticamente o espaço e sobretudo a complexidade de circuitos seqüenciais, sem falar dos custos.

PROPRIEDADES

Os diversos tipos de microprocessadores oferecidos no mercado podem ser classificados pelas suas propriedades, para efeito de seleção do tipo que melhor atende às características de uma dada aplicação, ou especificação de projeto, ou avaliação de desempenho relativo. Para descrever propriedades, explicitaremos o que se pode entender por:

• **Arquitetura** é o conjunto dos elementos e das regras de interdependência entre eles, que implementam a solução de determinado problema.

Assim se pode livremente falar de ARQUITETURA DE SISTEMAS, como referência a seus diversos dispositivos e técnicas de interligação. Se considerarmos cada dispositivo como um sistema em si, podemos novamente empregar o termo: arquitetura de uma unidade de disco, ou de uma CPU. E assim por diante, até chegar ao nível de componentes.

No caso dos componentes eletrônicos de larga escala de integração (LSI e VLSI), sua complexidade justifica ainda falar de MICROARQUITETURA. É pois analisando a microarquitetura de um microprocessador que explicitamos suas propriedades e deduzimos sua capacidade.

QUANTO À CAPACIDADE DE MEMÓRIA PRINCIPAL

É importante saber qual é a capacidade máxima de memória diretamente endereçável. Este é o prâmetro que limita o tamanho dos programas e suas áreas de dados. A propriedade da microarquitetura que dilata a capacidade máxima de endereçamento é o número de bits da **Barra de Endereços**, que é o número de algarismos binários destinados a acessar posições de memória. Exemplos:

INTEL 8080 e 8085, MOTOROLA 6800, MOSTEK 6502, ZILOG Z8002

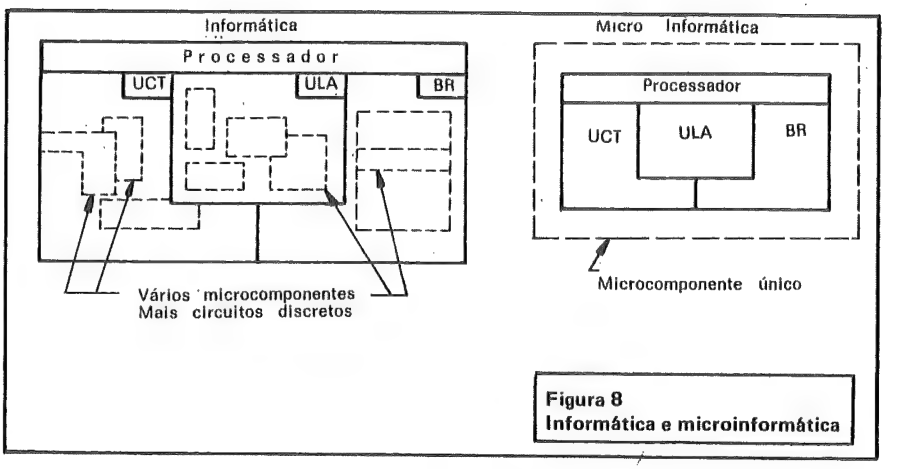


Figura 8
Informática e microinformática

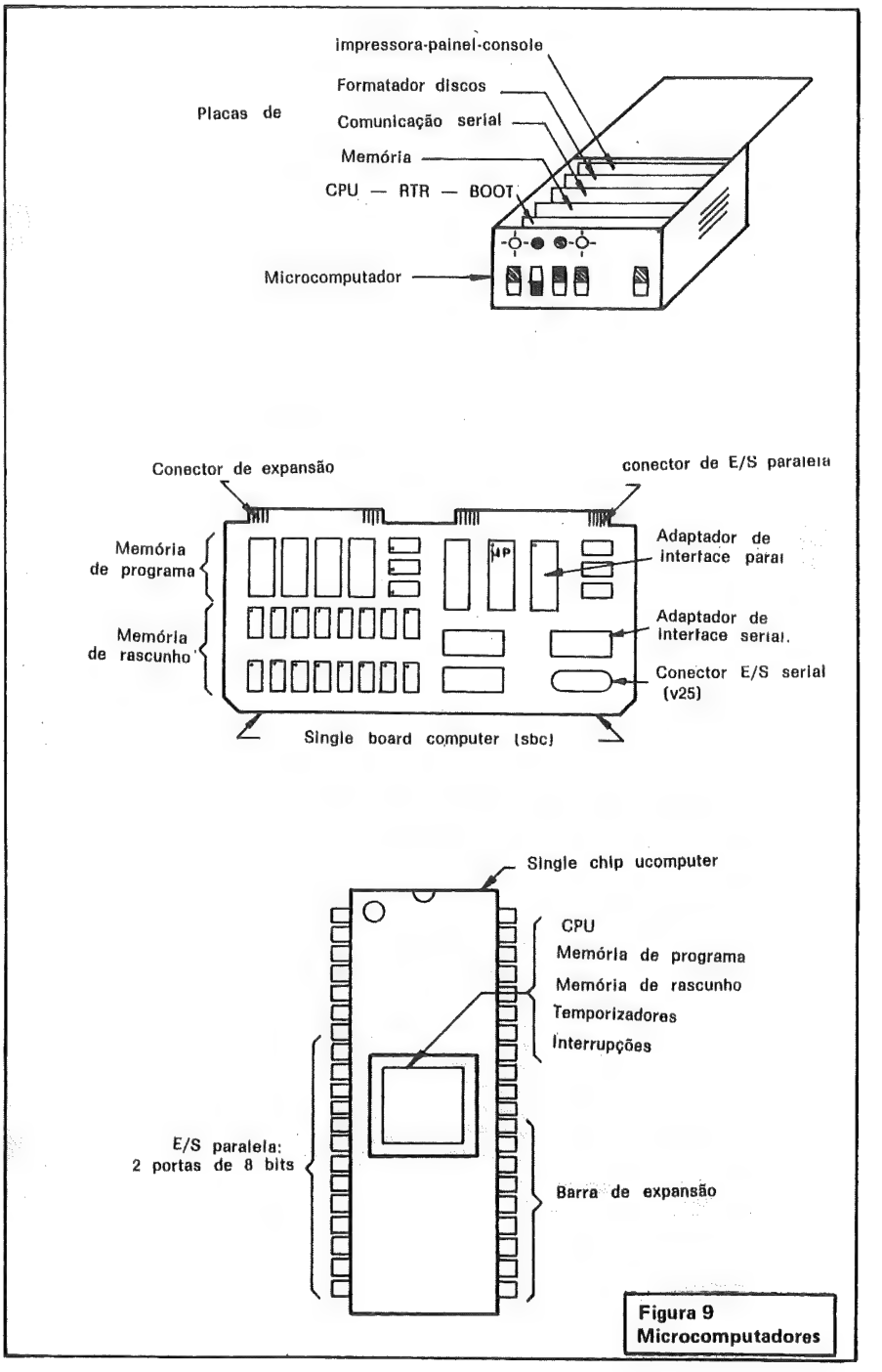
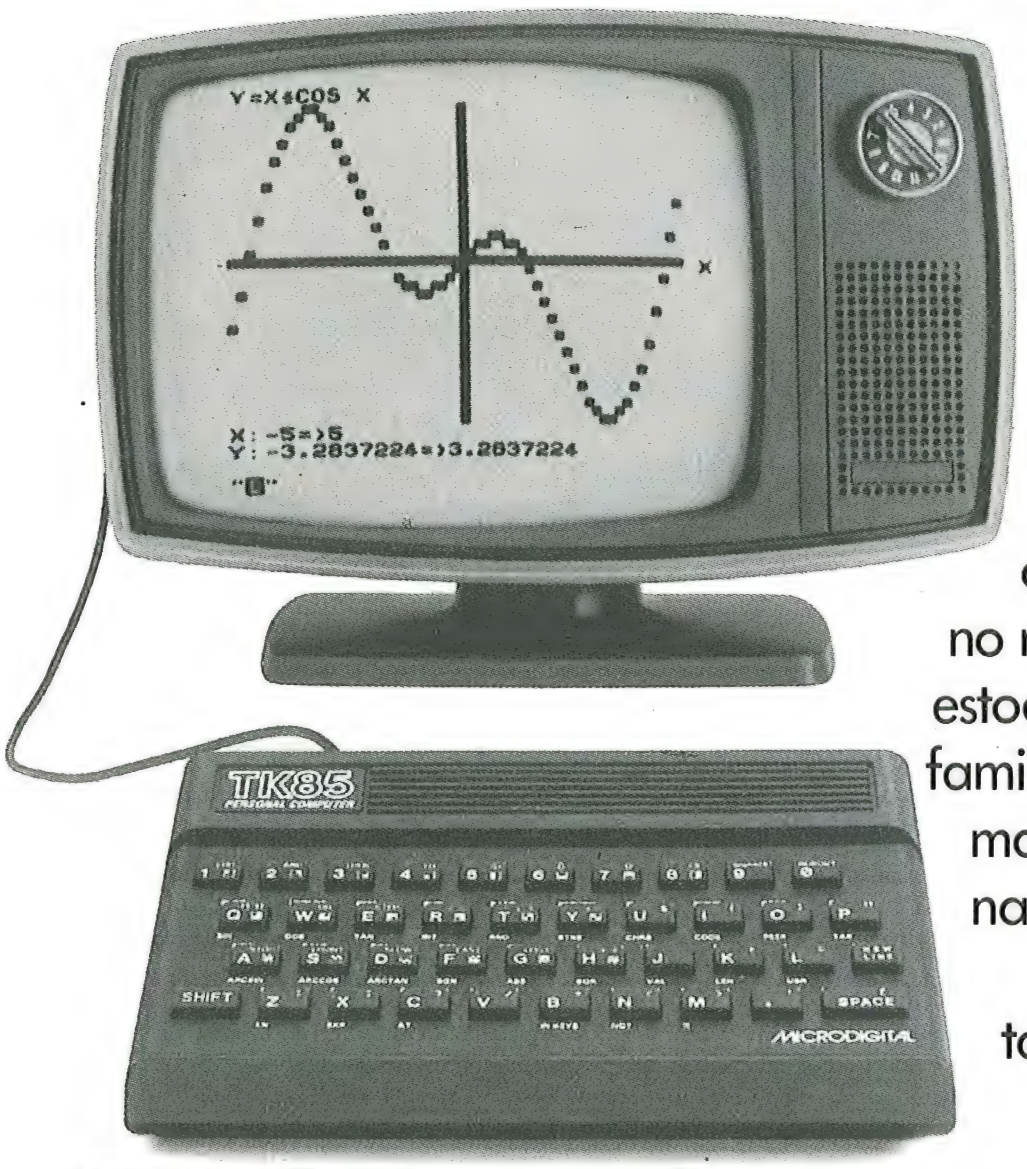


Figura 9
Microcomputadores

Nunca compre uma coisa que você não vai usar.



Leve logo um microcomputador TK 85, porque ele é realmente fácil de usar: já vem com manual de instruções, que ensina, em português claro, a linguagem Basic.

A partir daí, você pode preparar seus próprios programas ou utilizar as centenas de programas que já existem no mercado, para cadastrar clientes, controlar estoques, manter em ordem o orçamento familiar, fiscalizar a conta bancária, estudar matemática, estatística, jogar xadrez, guerra nas estrelas, e o que mais você puder imaginar.

E além disso tudo, o TK 85 tem também o preço mais acessível do mercado.

Peça uma demonstração.

TK 85, o micro que você pode usar.

MICRODIGITAL
computadores pessoais

Filiada a ABICOMP

FOX

unitron
a base de um sistema
inteligente

Quanto mais complexo for um sistema, mais sólida e confiável deve ser sua base. Quando você tem um micro da Unitron como princípio inteligente, você também tem a certeza de que o atendimento de suas necessidades em processamento de dados está assegurado. É a palavra de quem trabalha continuamente para oferecer uma tecnologia sempre atual ao usuário. É o que os fatos demonstram. Na sua categoria, o Unitron andou sempre na frente. Além de contar com uma infinidade de programas, testados e aprovados, e os mais diversificados acessórios de expansão – entre módulos, interfaces e periféricos –, o Unitron agora pode ser conectado, via telefone, a todas as redes existentes: Aruanda, Cirandão, Interdata, Cyber, Videotexto, CMA, etc. Ou, então, às redes particulares, acessando outros micros ou comunicando-se com computadores de grande porte, na função de terminal inteligente. Portanto, se você deseja um processamento de dados com qualidade, fale com nossos revendedores autorizados. Para cada caso, uma solução inteligente. Do princípio ao fim.

unitron

Computadores

CAIXA POSTAL 14.127 – SÃO PAULO – SP – TELEX (011) 32003 UEIC BR

possuem barra de endereços de 16 bits, logo:

216 = 65.536 = 64kb

INTEL 8086: 1Mb (com segmentação)

ZILOG Z8001: 8Mb (com segmentação)

MOTOROLA MC 68000: 16Mb (sem "overhead" de segmentação)

QUANTO AO PASSO DE TRANSFERÊNCIA

A informação do CICLO DE MEMÓRIA não é suficiente para avaliação da taxa de transferência. É preciso saber quantos bits são transferidos em cada ciclo. A **Barra de Dados** é a condutora dos bits de informação: o número de bits que a compõem é o passo de transferência de dados. Exemplos:

I 8080, I 8085, MC 6800, MK 6502, Z-80: 8 bits.

I 8086, Z 80801, Z 80802, MC 68000: 16 bits.

As máquinas de passo de 16 bits são também aptas a transferir 8 bits de cada vez, dependendo da instrução usada.

QUANTO AO NÚMERO DE BITS DOS OPERADORES DA ALU

Esta é a propriedade da microarquitetura mais usada para classificar microprocessadores. Diz-se que um dado microprocessador é de **n** bits quando sua unidade lógica e aritmética admite operandos de **n** bits.

Geralmente este número é o mesmo da BARRA DE DADOS, mas há exceções. Todos os micros listados no exemplo anterior, tem ALUs de passo igual ao da BARRA DE DADOS, **exceto** o MC 68000, cuja microarquitetura é de 32 bits. O passo da ALU determina também o tamanho dos registradores internos.

QUANTO À VERSATILIDADE DE PROCESSAMENTO

Quanto maior o número de funções de transferência implementadas num sistema programável, maior sua versatilidade. Nos processadores (micro ou não), a função de transferência é selecionada através de entidades denominadas INSTRUÇÕES, conforme já vimos.

Ao conjunto total das funções de transferência, corresponde o CONJUNTO DE INSTRUÇÕES do (micro) processador. Evidentemente, a versatilidade deve ser avaliada criteriosamente com o exame de cada instrução do conjunto, pois é comum encontrar várias instruções do mesmo tipo, ou seja, selecionando funções praticamente idênticas. Veja um exemplo retirado do microprocessador I 8085: as instruções ADD r, ADD M e ADI dado, adicionam o conteúdo do acumulador a um operando indicado na própria instrução (r = registrador interno, M = memória, dado).

A única diferença entre as três está na indicação do segundo operando. Conceitualmente, são três instruções idênticas, com diferentes modos de endereçamento (direto a registro, indexado e imediato) do segundo operando.

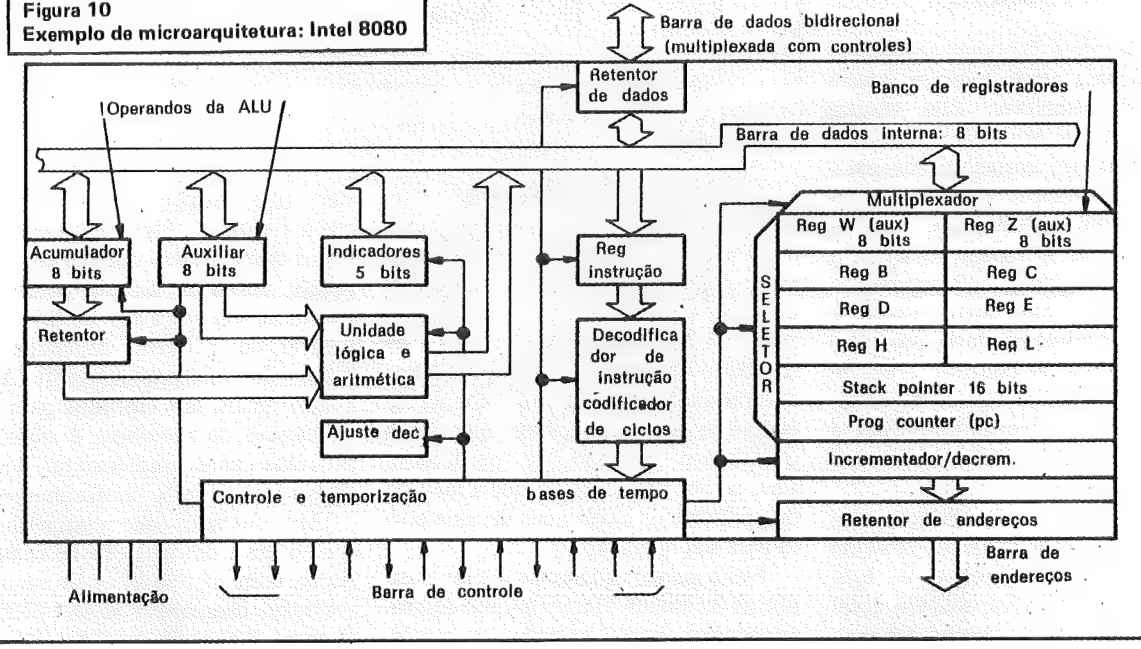
Não basta, portanto, saber o número de instruções constantes no CONJUNTO DE INSTRUÇÕES. É mais interessante levantar o número de tipos de instruções. Exemplos: I 8085 (que acabamos de exemplificar) possui mais de 100 instruções de aproximadamente 20 tipos diferentes. Z 8000 com 414 instruções de 110 tipos distintos.

Note nestes exemplos que o I 8085 tem bem menos instruções que o Z 8000, diferença explicável, pois o primeiro é um processador de 8 bits (ALU e passo) e o outro de 16 bits (ALU e passo). Passos maiores implicam em microarquitetura com maior capacidade nos seus registradores internos e lógica de controle, admitindo mais instruções com maior poder em cada uma.

Mesmo em microprocessadores de mesmo nível de microarquitetura, encontramos grandes diferenças na versatilidade.

O Z 80 e o I 8085 exemplificam este aspecto: o primeiro apresenta, entre outras vantagens, instruções de

Figura 10
Exemplo de microarquitetura: Intel 8080



controle automático de ciclos (DO-WHILE), e conversão HEXADECIMAL para ASCII.

Basicamente, o que se procura ao estudar o conjunto de instruções para avaliação de versatilidade são propriedades que facilitem sem desotimizar:

a) operações aritméticas e lógicas, que dependem diretamente da capacidade da ALU. Funções implementadas diretamente na ALU apresentam três vantagens: rapidez em relação às implementações por programa; economia de memória, pois rotinas são substituídas por instruções, esforço de programação. Em geral os micros de 8 bits só apresentam soma e subtração na parte aritmética e na parte lógica funções E, OU (inclusivo e exclusivo) e complementação. Tudo com precisão de 1 byte. Os micros de 16 bits já apresentam multiplicação e divisão com precisão de 2 bytes e alguns de 4 bytes, com aritmética inteira ajustável a decimal.

b) operações com transporte de dados entre o processador e a memória principal. São os MODOS DE ENDEREÇAMENTO, responsáveis pela eficiência da programação, na manipulação de estruturas de dados.

c) facilidades diversas como controle de ciclos (DO-WHILE, DO-UNTIL), conversões de códigos (BINÁRIO, DECIMAL, ASCII, BCD), operações com cadeias de caracteres, recursividade, transferências em blocos, manipulação de bits, etc.

Figura 11
Exemplo Assembler I 8080 e I 8085

Numeração automática de linhas

* — Linha de comentário

```
                0510 *****************************************************
                0515 *                                                   *
                0520 *   JOY-STICK                                       *
                0525 *                                                   *
                0530 *      ROTINA BASICA                                *
                0535 *                                                   *
                0540 *           - JOYLE => LE POSICAO XY DO JOY-STICK   *
                0545 *                                                   *
                0550 *****************************************************
                0555 *
142D            0560          ORG  *
                0565 *
                0570 *** LE POSICAO DO JOY-STICK NO CONVERSOR A/D ********
                0575 *        XB <= LEITURA DO CONV X
                0580 *        XC <= LEITURA DO CONV Y
                0585 *        DESTROI XA E FLAGS
                0590 *
142D 3E0C       0595 JOYLE    MVI  A,INICAD1   ENVIA PULSO
142F D3E7       0600          OUT  EC551       PARA INICIAR
1431 3E0D       0605          MVI  A,INICAD0   CONVERSAO A/D
1433 D3E7       0610          OUT  EC551       PARA OS CONVERSORES X E Y
                0615 *
1435 DBE5       0620 JOYLE1   IN   PC551       AGUARDA
1437 E608       0625          ANI  08          TERMINO DA
1439 C23514     0630          JNZ  JOYLE1      CONVERSAO
                0635 *
143C 3E09       0640          MVI  A,HABADX    HABILITA CONVERSOR X
143E D3E7       0645          OUT  EC551       PARA LEITURA
1440 DBE4       0650          IN   PA551       LE E ARMAZENA
1442 47         0655          MOV  B,A         CONTEUDO EM XB
1443 3E08       0660          MVI  A,DESADX    DESABILITA LEITURA DO
1445 D3E7       0665          OUT  EC551       CONV X (VOLTA A ALTA-IMP.)
                0670 *
1447 3E0B       0675          MVI  A,HABADY    HABILITA CONVERSOR Y
1449 D3E7       0680          OUT  EC551       PARA LEITURA
144B DBE4       0685          IN   PA551       LE E ARMAZENA
144D 4F         0690          MOV  C,A         CONTEUDO EM XC
144E 3E0A       0695          MVI  A,DESADY    DESABILITA LEITURA DO
1450 D3E7       0700          OUT  EC551       CONV Y (VOLTA A ALTA-IMP.)
                0705 *
1452 C9         0710          RET
                0715 *
                0720 *** EQU'S ***************************************
                0725 *
000D            0730 INICAD0  EQU  0D          SETA BIT P/ INICIAR CONVERSA
000C            0735 INICAD1  EQU  0C          RESETA BIT P/ INICIO DE CONV
0009            0740 HABADX   EQU  09          HABILITA CONVERSAO X
000B            0745 HABADY   EQU  0B          HABILITA CONVERSAO Y
0008            0750 DESADX   EQU  08          DESABIL CONVERSAO X
000A            0755 DESADY   EQU  0A          DESABIL CONVERSAO Y
                0760 *
```

Comentários

Operando (s) — se houver

Instrução

Rótulo para refer. endereçamento direto

Linguagem de máquina — instrução de 1, 2 ou 3 bytes

Endereços em representação hexadecimal

QUANTO À CAPACIDADE DE ENTRADA/SAÍDA

As portas de E/S podem ser sempre projetadas para atuar como se fossem posições de memória. Esta técnica recebe o nome de MEMORY MAPPED I/O. No entanto, certas inconveniências desta técnica são superadas quando o microprocessador oferece facilidades para a separação física e lógica entre a região de endereçamento da memória e das portas de E/S.

A microarquitetura deve, para tanto, prover sinais de controle e instruções que diferenciem acessos à memória dos acessos a periféricos. Esta é a técnica chamada SEPARATE I/O ou I/O MAPPED I/O.

É verdade que a imaginação dos projetistas consegue mapear memória no espaço de endereçamento de E/S, expandindo assim a capacidade total de endereçamento do sistema (chama-se I/O MAPPED MEMORY). Mas esta técnica limita a versatilidade do processamento por um motivo simples: as instruções de E/S não possuem o poder das de memória; restringem-se apenas à administração ou envio de um dado (IN e OUT). Um parâmetro determinante é o número máximo de portas de E/S endereçáveis, que limita o número máximo de periféricos. Leve-se em consideração que a maioria dos periféricos exige mais de uma porta para sua operação.

Neste ponto, queremos deixar claro que as facilidade oferecidas pela microarquitetura para operações de periféricos não chegam ao nível de distinguí-los entre si. Eles continuam sendo vistos como portas que operam basicamente como se fossem posições de memória endereçáveis a byte ou a palavra de 2 bytes, dependendo do passo do microprocessador. Apenas alguns sinais são emitidos para inibir os bancos de memória, e outros são providos para implementar técnicas específicas de E/S, conforme veremos.

Aos circuitos que compatibilizam a eletrônica de operação de cada periférico com os requisitos do microprocessador, chamamos de INTERFACE. O meio físico de comunicação entre

Direto aos Livros.
A Leitura Indireta já Era!

APOSTILAS, CÓPIAS REPROGRÁFICAS, ETC.

(1) **G. Gane/T. Sarson — Análise Estruturada de Sistemas**
1ª edição — 270 páginas — Cr$ 13.200,00
Apresenta um conjunto evoluído de técnicas e instrumentos surgidos do sucesso da programação e controle empresarial.

(2) **P. Bianchi/M. Bezerra — Microcomputadores — Arquitetura — Projeto — Programação**
1ª edição — 265 páginas — Cr$ 12.300,00
Profissionais, estudantes e os entusiastas do computador pessoal, encontrarão aqui, um meio de satisfazer a sua ânsia de aprofundamento.

(3) **A. Von Staa — Engenharia de Programas**
1ª edição — 300 páginas — Cr$ 14.700,00
O objetivo é apresentar uma seqüência harmônica e consistente das técnicas e métodos que visam à construção econômica de programas (SOFTWARE), com elevado nível de qualidade.

(4) **R. C. Pereira — Computes, Grilo! Computador Para Todas as Idades**
2ª edição — 112 páginas — Cr$ 6.500,00
Livro para crianças a partir de 10 anos que ensina a operar computadores pessoais. Fartamente ilustrado, acompanhando uma régua de fluxograma.

(5) **J. M. da Silva/W. E. Heibel — CIRCALC-VISICALC-PROCALC**
1ª edição — 172 páginas — Cr$ 8.000,00
A simplicidade de uso, a lógica de funcionamento clara e de fácil entendimento e a não exigência de experiência anterior, fazem dos denominados "Formulários Eletrônicos" eficientes instrumentos de primeiro contato entre o usuário e o computador.

(6) **F. Boratto — BASIC Para Engenheiros e Cientistas**
1ª edição — 128 páginas — Cr$ 7.500,00
Obra única em nosso idioma, contendo aplicações científicas em linguagem BASIC.

(7) **A. de Vasconcellos/C. Szerman — O Centro de Processamento de Dados**
1ª edição — 112 páginas — Cr$ 5.600,00
Esta publicação oferece todas as informações necessárias sobre o funcionamento de um CPD.

(8) **J. A. Borges — BASIC — Aplicações Comerciais**
2ª edição — 208 páginas — Cr$ 9.900,00
A abordagem geral do livro se baseia em exemplos voltados para aplicações do tipo gerencial.

(9) **L. M. Tarouco — Redes de Comunicação de Dados**
3ª edição — 204 páginas — Cr$ 9.200,00
Nesta 3ª edição foram incluídos novos tópicos descrevendo as inovações tecnológicas e as novidades no panorama brasileiro, tanto em termos de novos serviços de comunicação de dados, como regulamentações e tarifas vigentes.

(10) **J. B. Bosch — COBOL — Fundamentos e Aplicações**
1ª edição — 192 páginas — Cr$ 9.200,00
O livro aborda os comandos básicos e diferentes ferramentas do COBOL, tais como: Report Writer, Scort Intrínseco, Debug, pesquisas e tabelas, métodos de acesso, etc.

(11) **R. Prates — Cartão de Referência — CP-500**
1ª edição — Sanfonado — Cr$ 5.800,00
Abrange todos os comandos dos TRSDOS e todas as declarações utilizadas no BASIC. Qualquer informação pode ser utilizada rapidamente, evitando consumo de tempo e esforço. Instrumento de grande valia na programação e no uso do microcomputador CP-500 da PROLOGICA.

Não perca tempo.
Preencha o cupom anexo e receba o pedido em sua casa ou escritório, sem qualquer despesa extra. O envio dos livros será feito pelo serviço de reembolso postal.
Pague apenas quando for retirar o seu pedido.
Se preferir remeta cheque nominal a LTC — Livros Técnicos e Científicos Editora S.A. — Caixa Postal 3954 — 20001 — Rio de Janeiro — RJ — Preços válidos até 30/10/84.

LIVROS TÉCNICOS E CIENTÍFICOS EDITORA S.A.

A — 10

Remeta o seu pedido para:
20001 — RIO DE JANEIRO — RJ
Caixa Postal: 3954
Marque com um X o(s) livro(s) desejado(s)

1	2	3	4	5	6	7	8	9	10	11

Nome (pessoa física ou jurídica).....
CGC/CPF ..... Inscr. Est. .....
Endereço .....
CEP ..... Cidade ..... Estado .....
Assinatura ..... Data ...../...../.....

CPU e periférico pode ser SERIAL ou PARALELO.

As interfaces comunicam-se com a CPU sempre em paralelo, em passo igual ao do microprocessador. Do lado do periféfico, a interface executa as transferências de acordo com a natureza e a eletrônica de operação do dispositivo para o qual foi desenvolvida. Encarrega-se não só da comunicação dos dados, como também dos protocolos a nível de circuito ("hand-shake") e controle das transmissões

Fica assim transparente à micro-arquitetura, qual tipo de periférico usado, como exceção, declaremos o caso do microprocessador I 8085, que possui E/S serial direta para aplicações restritas (compatível TTL, sem "hand-shake").

O acima exposto é suficiente para implementação de técnicas de "POOLING" para gerência de periféricos. Outras facilidades são incluídas na microarquitetura para uso de técnicas de INTERRUPÇÃO e ACESSO DIRETO A MEMÓRIA (DMA), possibilitando a implementação de canais de E/S independentes ou por roubo de ciclo.

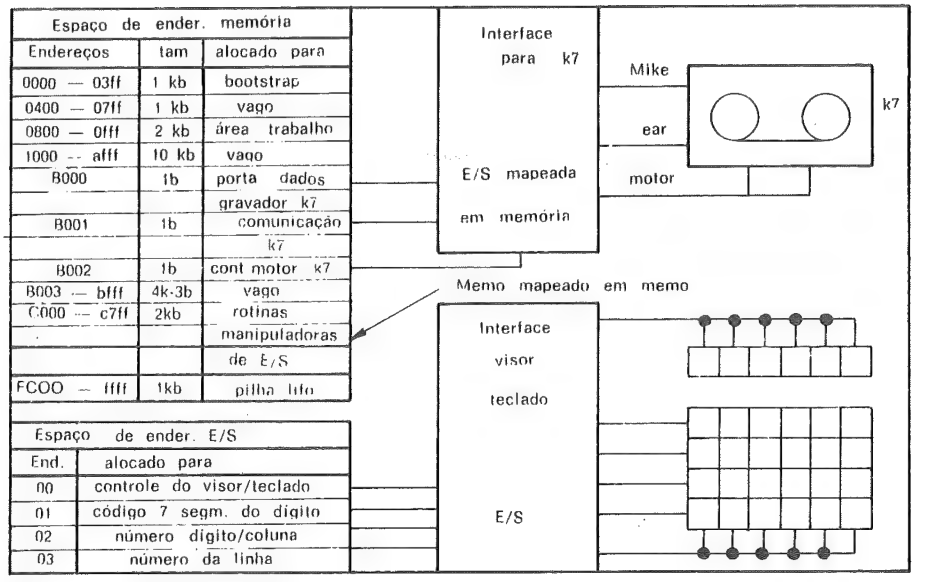


Figura 12
Mapeamento de endereços

UNIDADES DE APOIO

Sendo o microprocessador um componente eletrônico, necessita, para funcionar, do apoio de um circuito externo a ele, com função de transformá-lo num *microcomputador.* É a própria implementação da micro-arquitetura, ou simplesmente arquitetura do sistema, no que se refere ao que chamamos de *hardware.*

Também neste campo a microeletrônica tem contribuído decisivamente na simplificação dos circuitos, oferecendo componentes de grande capacidade lógica, devotados a fins específicos e com comportamento programável pelo processador. Vejamos os principais:

MEMÓRIAS

Circuitos integrados de memória se proliferam enormemente, sendo oferecidos com as mais diversas características de organização interna, potência consumida, rapidez de acesso, volatilidade, etc. Mais comumente encontramos na literatura os termos ROM e RAM para designar respectivamente Road Only Memory e Random Access Memory. Cada um com uma série de subgrupos.

As ROM's caracterizam pela não volatilidade das informações armazenadas quando da falta de alimentação elétrica do sistema, o que não ocorre com as RAM's. Em compensação estas admitem tanto leitura como gravação dinâmica, enquanto aquelas são de apenas leitura, uma vez que as informações são gravadas durante o processo de fabricação, sob especificação do cliente.

Assim as RAM's são usadas como "rascunho", ou seja, área de trabalho na qual são mantidos dados e até programas que uma vez utilizados são *randomicamente* substituídos. As ROM's são usadas para manter programas críticos, tais como *bootstrap,* núcleos de sistemas operacionais e rotinas de uso intenso.

Em busca do componente de memória ideal, capaz de unir as vantagens de não volatilidade das ROM's com a liberdade de gravação randômica das RAM's, a indústria tem dado alguns passos importantes.

As PROM's (Programable ROM) são ROM's virgens, cujas informações são gravadas pelo próprio usuário, via circuitos especiais definidos pelo fabricante. Aceitam apenas uma vez a gravação.

As EPROM's (Eraesable PROM) são PROM's que aceitam mais de

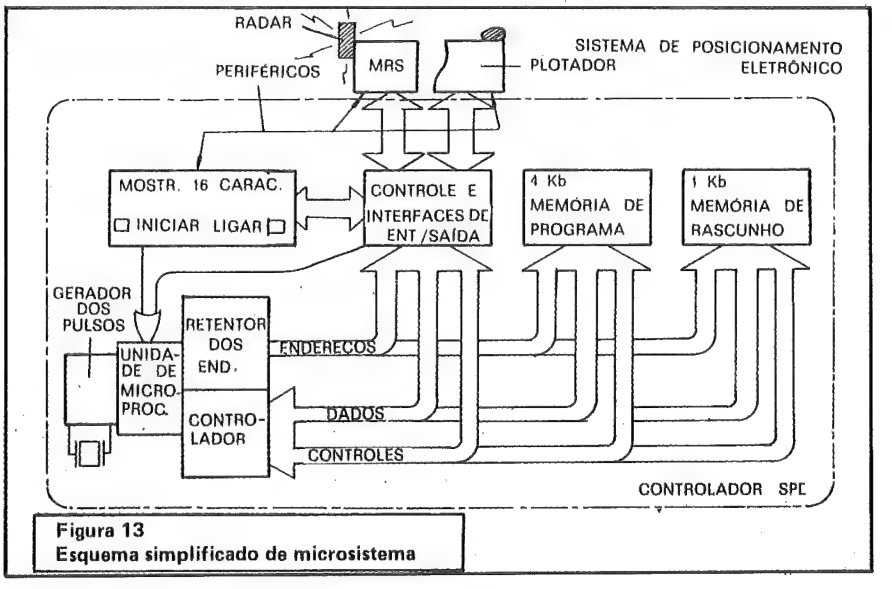


Figura 13
Esquema simplificado de microsistema

Escolha aqui o seu software

Para TK 2000 COLOR

- NIGHT MISSION - Night Mission e UFO.
- ARANHAS - Bug Attack e Night Crawler.
- TAXMAN - Taxman e Órgão.
- RESGATE SUICIDA - Resgate e Suicídio.
- ELIMINATOR - Eliminator e Dung Beetles.
- PLANO ZERO - Ceiling Zero, Go-Moku e Batalha Naval.
- GOBBLER - O melhor jogo já feito para micros.
- SPACE ATTACK - Efeitos sonoros fantásticos.
- DESAFIO FATAL - Une inteligência e ação.
- CORRIDA MALUCA - Uma verdadeira loucura.
- CONTRA-ATAQUE - Destruir a fortaleza é a meta.
- OPERAÇÃO PERIGO - Sensacional jogo animado.

Para TK 85 e CP 200

- PASSAGEM PARA O INFINITO - Um grande jogo.
- MIDWAY - Uma batalha no Atlântico.
- VISITA AO CASSINO - Quatro jogos de azar.
- JORNADA NAS ESTRELAS - Uma batalha espacial.
- SEGUNDA DIMENSÃO - Dois jogos de ação.
- 10 JOGOS EXCITANTES PARA 1 K - Para os pequenos.
- TERCEIRA DIMENSÃO - Terceira Dimensão, Fungalóides II, Super Penetrador I e Super Penetrador II.
- VELHO OESTE — Night Gunner, Compositor, Duelo Mortal e Túmulo Azteca.
- VIAGEM GALÁCTICA - Galáctica II, Invasores II e Alerta Vermelho.
- CONTATOS IMEDIATOS - Super Combate, Mother Ship, Força Invasora, Asteroide II.
- A GRANDE SABOTAGEM - Sabotagem, Maze Man, Super Demolidor, Meteoro-Super Nova.
- MR. KONG - Krazy Kong, Snakebite, Centopéia.
- ROT - 3 - High Speed, Super Scroll, CLS Espiral, Rotina Tape e Faster.
- SKBUG - Disassembler e monitor I.m.
- Z-80 MONITOR ASSEMBLER - O companheiro de SKBUG.
- GRAND PRIX - Fórmula 1 e Turbo.
- BOEING 787 - Super Simulador 787, Super Etandard e Space Raid.

Fita virgem especial para microcomputadores LD-20. 13 dB de ganho em gravações.

LIVRO LANÇAMENTO: "OS 40 MELHORES JOGOS PARA SEU TK/CP200 E SINCLAIR"

SE VOCÊ NÃO ENCONTRAR EM SUA LOJA, ESCREVA-NOS!

Qualidade Garantida

A SOFTKRISTIAN tem o melhor e mais moderno processo de industrialização, pioneiro no setor. Suas gravações são realizadas por modernas máquinas profissionais, eliminando de vez qualquer problema de carregamento. Suas fitas contém o exclusivo sistema ARS - Azimuth Regulating System, que permite a regulagem de seu gravador, mantendo o padrão internacional para fitas cassetes.

As fitas SOFTKRISTIAN tem garantia permanente contra qualquer eventual defeito de gravação, e são apresentadas em embalagens lacradas, para sua maior segurança; sem falar em sua apresentação gráfica, dando um acabamento superior ao produto.

Todos estes motivos fazem com que, hoje, a SOFTKRISTIAN seja líder absoluta em produção de software no mercado brasileiro de informática. SOFTKRISTIAN, o seu melhor software.

Revendedores Softkristian

ALAGOAS: MACEIÓ: Expoente - AMAZONAS: MANAUS: Cap / Imp. Oliveira / Mesbla - BAHIA: SALVADOR: Mesbla / Officina - FEIRA DE SANTANA: Micrológica - CEARÁ: FORTALEZA: Mesbla / Microcenter / Systematic - DISTRITO FEDERAL: BRASÍLIA: Compushow - ESPÍRITO SANTO: VITÓRIA: Mesbla - GOIÁS: GOIÂNIA: Mesbla / Radelgo - MATO GROSSO: CAMPO GRANDE: DRL - MINAS GERAIS: BELO HORIZONTE: Computronix / Lojas Retes / Mesbla / Mikro Informática / Microshop - IPATINGA: Micro e Vídeo Eletrônico - JUIZ DE FORA: Exitus / Microbyte - LAVRAS: Zog - MIRACEMA: Leonardo Chamarelli - MURIAÉ: Regis Studio - POÇOS DE CALDA - Micropoços - TIMÓTEO: Micro e Vídeo Rodason - UBERLÂNDIA: Blow-up / Work Shop - VIÇOSA: Micron - PARÁ: BELÉM: Ação Imagic / Compubel / Mesbla - PARAÍBA: CAMPINA GRANDE: João Tomé Filho 321-2555 - JOÃO PESSOA: Mesbla - PARANÁ: CURITIBA: Computique / Madison / Mesbla - LONDRINA: Compushop / Komputer Haus / Mesbla - MARINGÁ: Compucenter / Control - PERNAMBUCO: RECIFE: Mesbla / Nordata / Souza's Computer Center / Televídeo - RIO DE JANEIRO: RIO DE JANEIRO: Actvision / Artmec / Bel Bazar / Ciência Moderna / Computique / Datamicro / Eletronic / Eletrodata / Eldorado / Entrelivros / Fotomania / Josias Studio / Joy Game Club / L'artisan / Leo Foto Som / MCS Distribuidora / Mesbla / Micro Informática / Micromaq. / Micromint / Micro News / New Vídeo / Ótica Suíça / Poligames / Prisma / Rio Micro / Robotic / Seletronix / Simpro / Sinclair Place / Space Equipamentos / String / Tele Rio / W. Shock - CAMPOS: Recrileo - MARICÁ: Cedapro - NITERÓI: Mesbla / New Vídeo - NOVA FRIBURGO: Gachet - PETRÓPOLIS: Microtronics - VOLTA REDONDA: Mesbla / Proserv - RIO GRANDE DO SUL: PORTO ALEGRE: Advancing / Digital / Informatique / JH Santos / Mark Data / Mesbla / Microsis - CAXIAS DO SUL: Geremia / Nordemaq - IJUI: Von Eye - NOVO HAMBURGO: Micromega - SANTA MARIA: Índia Center - SANTA ROSA: Informática Dinâmica - SANTANA DO LIVRAMENTO: Eletrolândia - PELOTAS: Mesbla - SANTA CATARINA: FLORIANÓPOLIS: Supermicro Show - BRUSQUE: Renaux - CRICIÚMA: Compumatic - JOINVILLE: Computerville - SÃO PAULO: SÃO PAULO: Accordyne / Áudio / Benny Feira / Bucker / Cinótica / Comercial Sul América / Computerland / Computique / Datashop / Eletrônica Santana / Exatron / Filcril / Fotoleo / Fotoptica / G. Disk / Guedes / Imares / Livraria Poliedro / Livraria Sistema / Mappin / Mesbla / Micromania / Micro Process / Microshop / Multison / Plandata / Pro Informática / Sys Dez / Trimaq - BAURU: Micrológica - CAMPINAS: Mesbla / Tropical - MARÍLIA: Mesbla / Sipro - MOGI DAS CRUZES: Jutaro Tamura - MOGIGUAÇU: Tropical - MOGI MIRIM: Tropical - PIRACICABA: RC Microcomputadores / Tropical / Palma - RIBEIRÃO PRETO: Compusys / Memocards / Mesbla - SANTO ANDRE: Mesbla / Shopping Áudio e Vídeo - SÃO CARLOS: Queops - SANTOS: Ritz Cine Foto - SERGIPE: ARACAJÚ: Micromundo.

SoftKristian®

O seu melhor Software

Rua Gonzaga Bastos, 112 - CEP: 20541 Rio de Janeiro - RJ - Tel. PABX: (021) 268-8249

A.G. Comunicações Visuais Ltda.

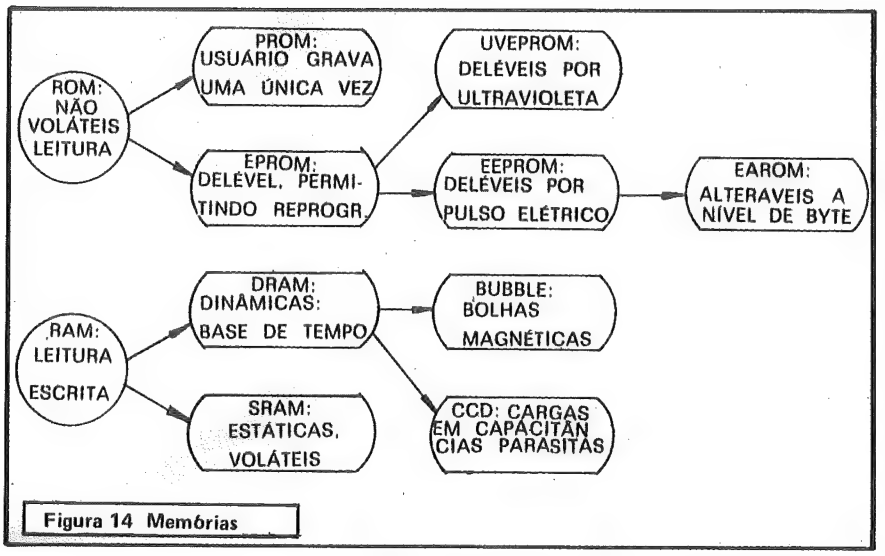


Figura 14 Memórias

uma gravação, contanto que sejam antes apagadas via exposição de tempo controlado, a raios ultravioletas ou a campos elétricos.

Tanto PROM's como EPROM's são gravadas fora do berço que lhes é destinado dentro da arquitetura do microcomputador.

Surgiram então as EEPROM's (Eletricaly Ereasable and Programable Rom) que podem ser alteradas (apagadas e reprogramadas) a nível de byte pelo próprio circuito da CPU, com algumas diferenças aceitáveis e com um tempo de gravação ainda bastante superior ao de leitura. São muito úteis em sistemas adaptativos ou quando informações devem ser alteradas esporadicamente.

As RAM's são genericamente diferenciadas entre ESTÁTICAS (SRAM) e DINÂMICAS (DRAM) para diferenciar as tecnologias que exigem uma base de tempo (clock) para operar.

No caso das DRAM's tipo BUBBLE MEMORY, a temporização é exigida para "tradução" do endereço que se deseja acessar pois possuem organização serial, o que lhes dá um baixo tempo médio de acesso, compensado por três fatores: baixo custo por bit, ocupação de reduzido espaço (1M bit em 10cm quadrados) e sobretudo não volatilidade.

No caso das DRAM's tipo CCD (Charge-Coupled Device) a base de tempo é exigida para *reforçar* periodicamente as informações armazenadas. Estas são representadas internamente por cargas elétricas que se escoam em uma ou duas centenas de milisegundos, período em que necessitam receber reforço. E mais: ao serem lidas são destruídas, o que obriga uma recomposição a cada leitura, feita aliás automaticamente pelo microcircuito do componente, tornando-se transparente ao usuário.

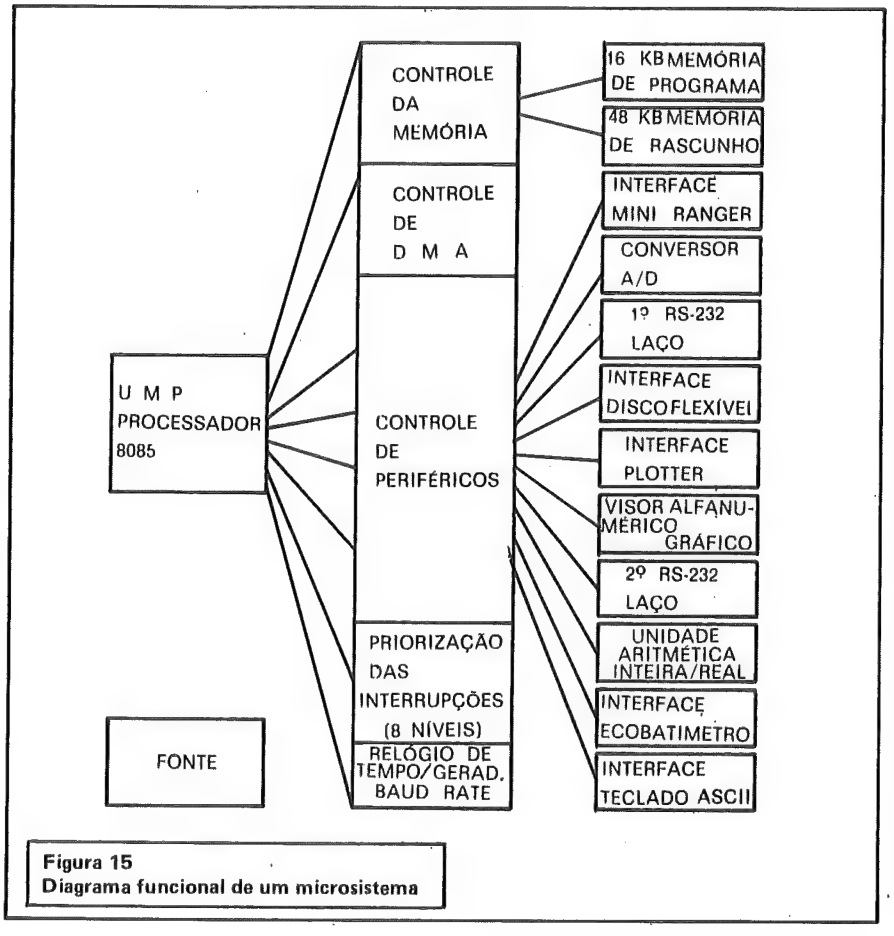


Figura 15
Diagrama funcional de um microsistema

INTERFACES E CONTROLADORES

São destinadas a:
– baratear custos de desenvolvimento e produção de sistema;
– aliviar o processador da maior parte do *overhead* de operação dos periféricos;
– facilitar os programadores das rotinas básicas de manipulação dos periféricos.

PIA's (Paralel Interface Adapter) ou PPI's (Programable Paralel Interface) são projetados para facilitar interfaces paralelas de propósito geral. Existem versões para barramentos padronizados.

USART's (Universal Synchronous / Asynchronous Receiver / Transmitter) para interfaces seriais tipo RS-232C, V-25, laço de corrente, com facilidades para conexão com modems. Existem versões que manipulam protocolos específicos (IBM – BORROUGHS CDC – etc).

DMA's (Direct Memory Access Controllers) são controladores de transferência de blocos MEMÓRIA – MEMÓRIA ou MEMÓRIA – PERIFÉRICO, em alta velocidade sem a concorrência do processador.

CONTROLADOR DE INTERRUPÇÕES vetoriza e estabelece prioridade para as interrupções dos diversos periféricos, expandindo a capacidade de

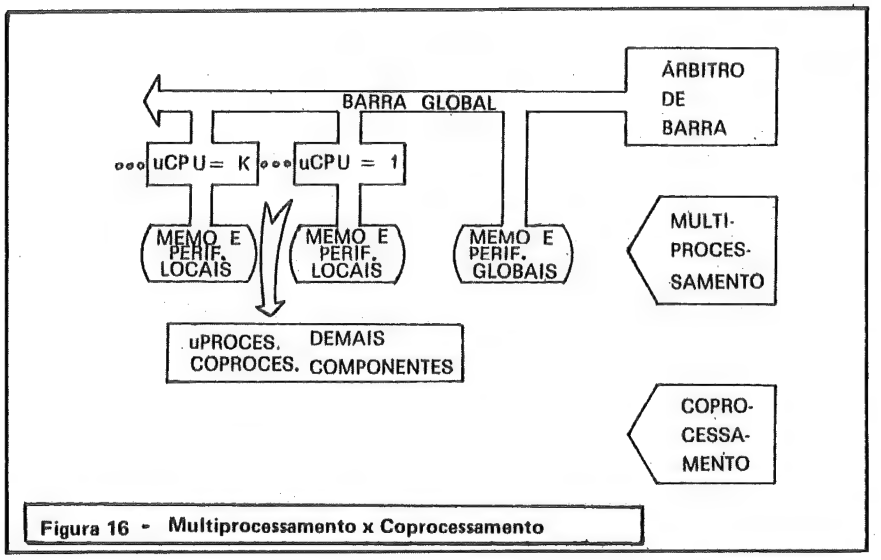


Figura 16 - Multiprocessamento x Coprocessamento

goritmos aritméticos.

CONTROLADORES DE DISCOS E FITAS reduzem em 70% os circuitos de controle direto dos mecanismos de armazenamento de massa em meios magnéticos.

CONTROLADORES DE VÍDEO fazem o mesmo com relação aos tubos de raios catódicos (CRT's) para exposição de informações alfanuméricas e gráficas.

COPROCESSADOR

Representa a mais recente inovação em microinformática, e está apenas "engatinhando", embora já existam alguns no mercado, como o I 8087 Numeric Data Processor.

vetoração do processador.

TIMERS cobrem diversas faixas de aplicação, podendo ser programados para interrupções periódicas do processador em diversos intervalos, ou mesmo limitando-se a manter um relógio/cronômetro de alta precisão, para consultas pelos programas.

ADC's e DAC's são conversores de sinais analógicos para digitais e digitais para analógicos, respectivamente, empregados com aplicações de controle e/ou monitoração de fenômenos não discretos. É o caso da monitoração de temperatura (de fornos, de dispositivos críticos), controle de rotação de motores elétricos etc. Existem outros integrados que convertem/revertem para digital fenômenos de freqüência, toque, etc.

APU's (Arithmetic Processing Units) são periféricos que implementam funções aritméticas, trigonométricas, exponenciais/logarítmicas, em várias precisões e admitindo operandos inteiros e de ponto flutuante. APU's diminuem consideravelmente o tempo de resposta em aplicações numericamente pesadas, aliviando o programador de penosas implementações de al-

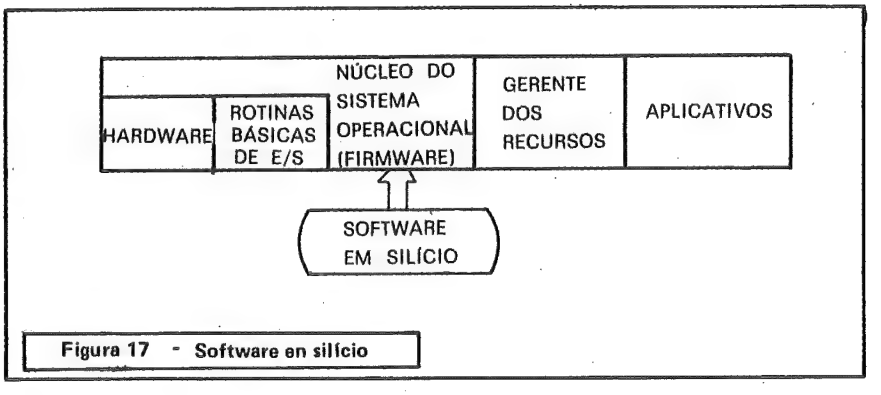


Figura 17 - Software en silício

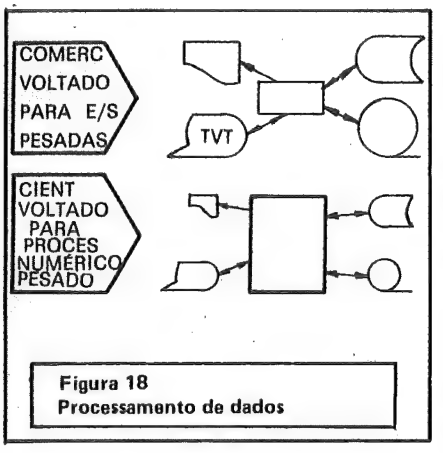


Figura 18
Processamento de dados

É importante não confundir com multiprocessamento, técnica também empregada largamente em arquiteturas baseadas em microprocessadores.

O coprocessador é um microprocessador otimizado para executar instruções de determinada índole (numéricas no caso do I 8087), ligado em paralelo com um microprocessador de propósito geral, monitorando todo o fluxo de instruções a este submetidas. Ao detectar uma que lhe cabe processar, emite aviso do fato e empreende sua tarefa, cujo término é também sinalizado.

Por Trás de Todo Bom Micro Existe Sempre Uma Boa Assistência Técnica
• Assistência Técnica para Micros Nacionais e Importados
MICROBIO
R. Almte. Tamandaré, 66 S/Loja 214 Esq. R. Catete – Rio de Janeiro Tel.: (021) 265-8294
Diariamente das 8 às 22 h e de plantão aos fins de semana.

SOFTWARE ENCAPSULADO

Com tantas facilidades de hardware, e com o aumento da capacidade de processamento dos microcomputadores, é de se esperar que o "gargalo" do desenvolvimento de sistemas em microinformática se desloque para o "reino do programador", vulgo software.

Diversas linguagens e sistemas operacionais vêm surgindo, novos ou adaptados de máquinas maiores, trazendo consigo alta sofisticação (multitarefas, multiprogramação, multiprocessamento, multiusuário, etc). Linguagens como o PASCAL, ADA, C e outras, são estruturadas; admitem alocação dinâmica de memória para estruturas de dados e rotinas *(overlays)*, otimizam acessos a arquivo, etc. Tudo vendido em discos flexíveis.

Mais recentemente, o mercado vem oferecendo em pastilhas de microcircuitos núcleos de sistemas operacionais (O. S. Kernel), executáveis independentemente de zona de endereços alocado, cobrindo 60% das necessidades de software no desenvolvimento de sistemas operacionais, sobretudo os de aplicação específica.

Exemplo: VRTX (pronuncia-se "vertex", ocupando 4Kb, adapta-se para I 8086, M 68000 e Z 8000. Oferece algumas dezenas de primitivas. O apelido desta técnica é "SILICON SOFTWARE".

APLICAÇÕES

Do leque de aplicações da microinformática podemos destacar alguns grupos, de conceituação aliás não muito bem definida, sobretudo porque destinam-se a fins específicos de porte pequeno, geralmente pluridisciplinares, graças à versatilidade que cada projeto pode dispor.

PROCESSAMENTO DE DADOS

É a aplicação mais corrente, empregada na automação de escritórios e pequenas firmas. No mercado nacional os microcomputadores estão quase todos situados neste grupo e baseiam-se nos micros mais difundidos.

COBRA 400	INTEL 8080
COBRA 400II	INTEL 8085
COBRA 305	ZILOG Z80
MICRO SCOPUS	ZILOG Z80
NOVADATA 86	INTEL 8086

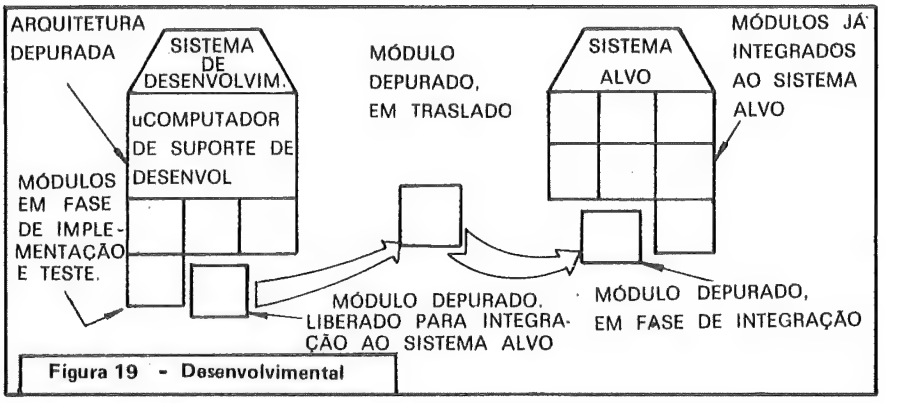


Figura 19 - Desenvolvimental

São oferecidos com pacotes aplicativos na área comercial e administrativa, economizando equipe de programação. Sua robustez não exige *overhead* de instalação.

Trazem sistemas operacionais em disco (geralmente flexíveis) e linguagem de programação de alto nível (geralmente BASIC e COBOL). Admitem impressoras seriais e paralelas, fita magnética e comunicação com sistemas maiores.

Infelizmente, ainda não contamos no Brasil com microcomputadores para aplicações científicas e de engenharia, que exigem larga capacidade de processamento numérico.

DESENVOLVIMENTAIS

São microsistemas projetados para suportar o desenvolvimento de novas aplicações, tanto em software, como em hardware. Apresentam facilidades de programação em alto nível, para implementação e teste das novas aplicações, que uma vez dadas como prontas são transferidas para o sistema alvo em linguagem de máquina, desobrigando-o de arcar com a sobrecarga de vastos sistemas operacionais, compiladores, interpretadores e ligadores de edição.

Também subsistemas novos de hardware são submetidos a testes no ambiente do desenvolvimental, o que garante a concentração do esforço sobre a porção em teste, com a certeza de uma arquitetura já depurada, o que não acontece quando o teste é feito sobre o sistema alvo, em fase de desenvolvimento. A GEPETO ELETRÔNICA com seu desenvolvimental SDC-G80/85, baseado no I 8085, é pioneira nacional neste ramo.

PROCESSAMENTO DE SINAIS

A ciência sonda fenômenos naturais a partir da análise de seus efeitos sobre nossos sentidos, e na falha ou imprecisão destes, usa aparelhos capazes de captar os sinais característicos de cada fenômeno, alguns dos quais o homem aprendeu a controlar e usar em seu próprio benefício (ou em malefício alheio – infelizmente).

Assim surgiu a comunicação por sinais de rádio, microondas, cabos, etc. Surgiram sistemas de medição precisa de distâncias aplicadas à metrologia científica (na legislação bra-

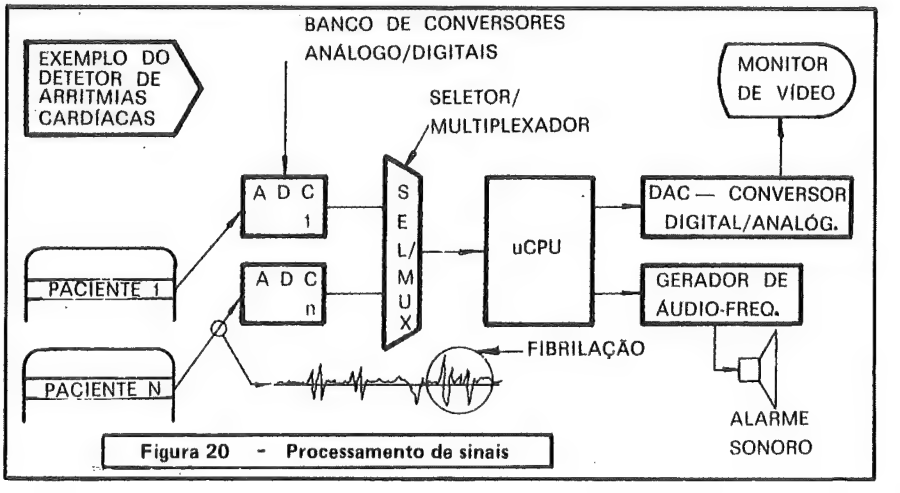


Figura 20 - Processamento de sinais

Como construir um satélite artificial sem sair da cadeira. Veja no Show de Informática 84 da IBM Brasil.

A IBM não pára de desenvolver novas tecnologias, em seus centros científicos e laboratórios de pesquisas em todo o mundo. Estas conquistas serão demonstradas em show de multivisão, com utilização de laser em seu stand na Feira de Informática.

Nele estarão, ao vivo, os mais novos produtos IBM, alguns dos quais pela primeira vez exibidos no país. Veja tudo o que a IBM oferece a você, seja qual for o seu campo de atividade profissional.

Entre as estrelas deste show de tecnologia, você vai ver o terminal IBM 5080, um avançado sistema gráfico, capaz de projetar, nos mínimos detalhes, desde o mais simples sólido até o mais complexo satélite espacial.

E as impressoras a laser, eletro-erosão e multi-velocidade.

E mais o sistema robótico IBM 7535, em ação, mostrando como se faz automação industrial.

Todas estas conquistas você vai ver em pleno funcionamento, demonstrando o que a IBM pode fazer pelo país e por uma vida melhor.

IBM

IBM Brasil

Show de tecnologia
para uma vida melhor.

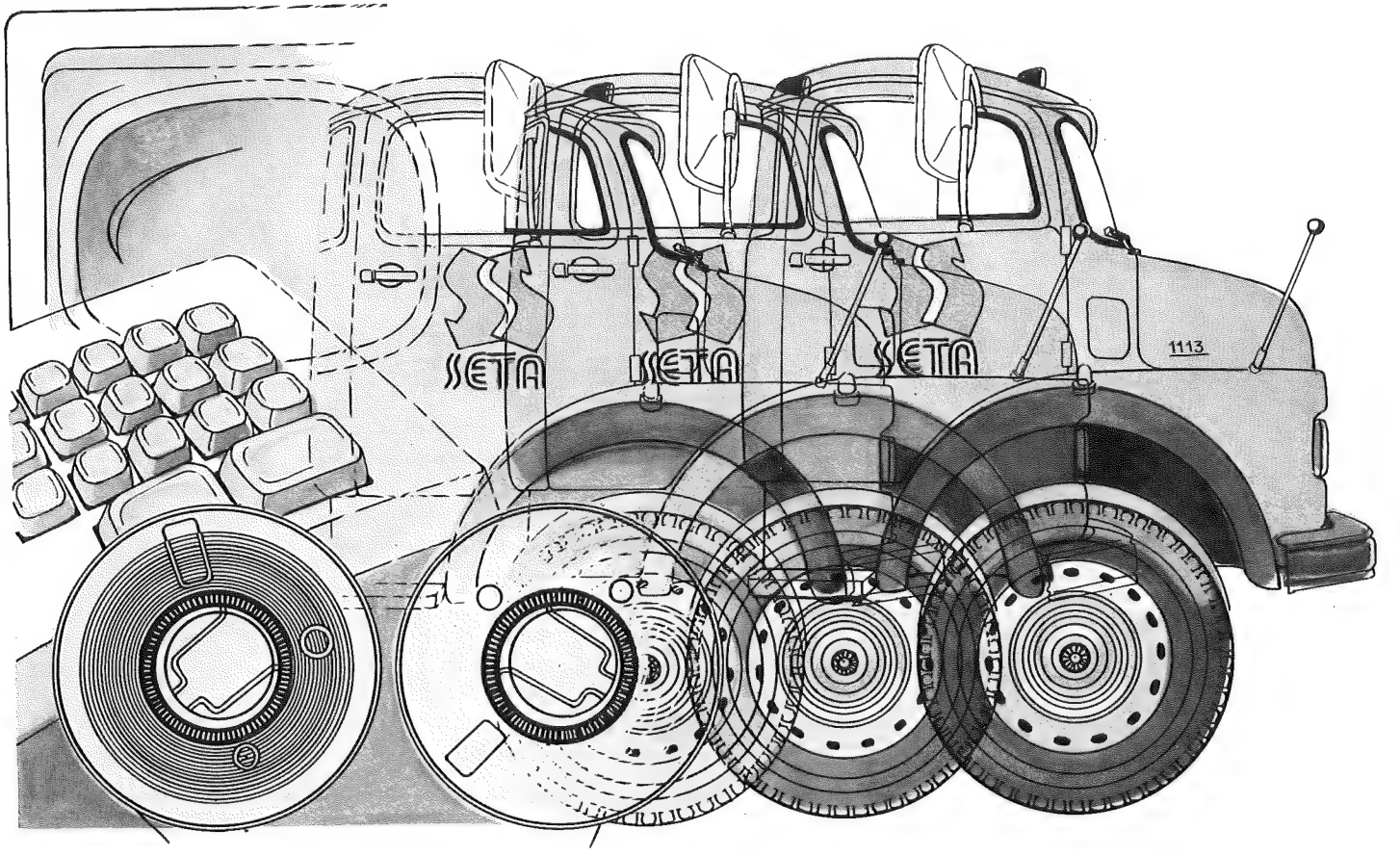
IV FEIRA INTERNACIONAL DE INFORMÁTICA - DE 5 A 11 DE NOVEMBRO - RIOCENTRO, RJ.

DENISON

SISTEMA SETA DE TRANSPORTE
DIVISÃO DE EQUIPAMENTOS SENSÍVEIS

Na IV FEIRA INTERNACIONAL DE INFORMÁTICA,
no Riocentro a Transportadora SETA estará
atendendo oficialmente a todos os expositores
em tempo integral normal ou em momentos de
urgência, durante todo o período do evento.
Visite o nosso estande localizado na entrada principal
do Pavilhão — Área Especial n.º 1.
Conheça, em uma exibição de audio-visual, a mais
moderna tecnologia do transporte de equipamentos sensíveis.

A INFORMÁTICA ESTRADA AFORA



EMPRESA DE TRANSPORTES SETA LTDA

RIO DE JANEIRO: TEL: PABX (021) 372-2969 - SÃO PAULO: TEL: PABX (011) 295-3122

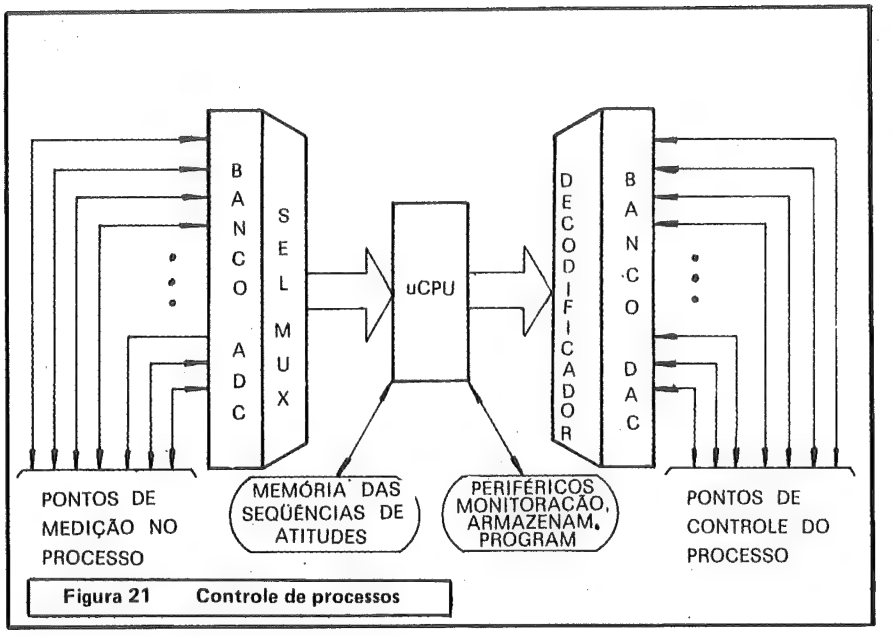


Figura 21 Controle de processos

sileira, 1(um) metro é definido como uma constante, vezes o comprimento de onda do XENÔNIO), topografia, navegação, etc.

Métodos não-invasivos de avaliação orgânica, como eletroencefalocardiograma, radiografia, ultra-sonografia, etc.

Medidas à distância, como temperatura de fornos industriais, velocidade de corpos (desde o radar do DETRAN até o radiotelescópio de Monte Palomar), migração de átomos em cristais sólidos, etc.

Os sistemas nesta área usam métodos numéricos de alta complexidade, geralmente atuando como filtro para extrair a informação desejada, isolando-a dentre as outras e eliminando ruídos e distorções produzidos por interferências de fenômenos espúrios atuantes no caminho percorrido pelo sinal. Um exemplo típico é o MICRODENSITÔMETRO do laboratório de Rádio-Astronomia da UFRJ, instalado no Observatório Nacional. Dada uma superfície pigmentada oticamente, são traçadas sobre ela as curvas de igual intensidade de pigmentação (isotônicas). Assim, radiofotografias, literalmente ininteligíveis, são processadas para isolamento de regiões (corpos ou grupamentos de corpos) de interesse do pesquisador.

CONTROLE DE PROCESSOS

Qualquer processo industrial, de laboratório ou de engenharia, é constituído por uma *seqüência ordenada de atitudes* que conduzem ao produto final. Controladores eletrônicos de processos são máquinas dotadas de grande capacidade de monitoração e atuação sobre as variáveis de um processo, cuja seqüência de atitudes encontra-se armazenada na memória em forma de programa, a fim de dar a elas execução automática.

CONVERSORES D/A (Digitais – Analógicos) convertem as variáveis do processo, geradas pelo processador na representação binária, em energia analógica aplicada sobre os pontos de controle.

CONVERSORES A/D (Análogo – Digitais) traduzem para binário os parâmetros de resposta do processo, que o sistema analisa para simples monitoração e eventuais tomadas de decisão.

O COBRA 700 é o pioneiro nacional em controle de processos. Usa tecnologia *bit-slice* e não é considerado microcomputador.

apple cursos e sistemas

CURSOS COM APOSTILA E AULAS PRÁTICAS

● **BASIC** para APPLE - Turmas de 12 alunos. Lógica de programação.

● **APPLESOFT/DOS 3.3** Turmas de 12 alunos Gráficos e arquivos em disco no APPLE

SISTEMAS APLICATIVOS PARA CP/M E APPLE

● **SISADE** - Administração Escolar Completa
● **SISMAT** - Monitor de aulas em micros (geral)
● **SISFRA** - Folha de Pagamento
● **SISCTB** - Contabilidade Geral
● **SISCPR** - Contas a Pagar e Receber
● **SISEST** - Controle de Estoque (vários tipos)
● **SISFAT** - Faturamentos (vários tipos)
● **SISCCM** - Controle Custos em Metalúrgicos
● **SISCCF** - Controle de Transportadoras
● **SISCAR** - Controle para Crediários em Geral
● **SISTIC** - Controle de títulos de Clubes
● **SISTAM** - Controle de tít. de Assist. Médica
● **SISACM** - Controle de fichas médicas

Executamos **Sistemas Aplicativos Gerais,** Criação de **Aberturas e Vinhetas** publicitárias por computador, sob encomenda.

MINIS, MICROS E PERIFÉRICOS

● **NOVADATA** - ND86MW/ND86E (supermicros e minis
● **TELSIST** - microcomputadores
● **TS1802** - monousuário com/sem winchester
● **TS1806** - multiusuário com super winchester
● **APPLES** - diversos, CPU, Drives, Vídeos, Placas
● **GLOBUS** - Impressoras de grande capacidade
● **ELGIN** - Impressoras matriciais
● **ELEBRA** - Impressoras e Modens
CMA - modens p/Videotexto e Cirandao
● **ITALMA** - Móveis para CPDs e escritórios

RUA SUZANO, 78 - tel: (011) 853-9457
01453 - JDIM PAULISTA - SÃO PAULO - SP.

SAFARI SOFT

SISTEMAS APLICATIVOS PARA CP/M E APPLE

* **SISADE** - Administração Escolar Completa
* **SISMAT** - Monitor de aulas em micros (geral)
* **SISFPA** - Folha de Pagamento
* **SISCTB** - Contabilidade Geral
* **SISCPR** - Contas a Pagar e Receber
* **SISEST** - Controle de Estoque (vários tipos)
* **SISFAT** - Faturamentos (vários tipos)
* **SISCCM** - Controle Custos em Metalúrgicas
* **SISCCF** - Controle de Transportadoras
* **SISCAR** - Controle para Crediários em Geral
* **SISTIC** - Controle de títulos de Clubes
* **SISTAM** - Controle de tít. de Assist. Médica
* **SISACM** - Controle de fichas médidas

* Executamos de **Aberturas Aplicativos Gerais,** Criação de **Aberturas e Vinhetas** publicitárias por computador, sob encomenda.

MINIS, MICROS E PERIFÉRICOS

* **NOVADATA** - ND86MW/ND86E (supermicros e minis)
* **TELSIST** - microcomputadores
TS1802 - monousuário com/sem winchester
TS1806, multiusuário com super winchester
* **APPLES** diversos, CPU, Drives, Vídeos, Placas
* **GLOBUS** - Impressoras de grande capacidade
* **ELGIN** - Impressoras matriciais
* **ELEBRA** - Impressoras e Modens
* **CMA** - modens p/Videotexto e Cirandao
* **ITALMA** - Móveis para CPDs e escritórios

RUA SANTA LUZIA, 799 gr 901
Tel.: (021) 399-4807 - tlx: 2123985 CBSR BR
20030 - RIO DE JANEIRO - RJ
RUA SUZANO, 78 - tel.: (011) 853-9457
01435 - JDIM PAULISTA - SÃO PAULO - SP.

O SISTEMA OPERACIONAL CP/M

Em virtude da proliferação, no mercado nacional, de um grande número de microcomputadores operando com o sistema operacinal CP/M ou similares (Micro SCOPUS, Sistema 700 da PROLÓGICA, Micro SHUMEC, SME 01 da P&D, Micro ITAUTEC, Alfa 2064/3000 da DISMAC e outros), resolvemos publicar esta série de artigos, com o intuito de propiciar aos usuários e pessoas que desenvolvem software para esses equipamentos, uma visão bastante detalhada do seu sistema operacional.

O que é um sistema operacional? Em geral, um sistema operacional é um programa ou conjunto de programas relacionados, cujo objetivo é agir como interface entre o usuário ou programa de aplicação e o hardware da máquina.

Na realidade, quando ordenamos a um microcomputador que carregue um programa, liste um programa ou arquive no vídeo ou impressora, mostre o diretório de um disco, salve um programa ou texto em disco, etc, quem executa esses comandos é o **sistema operacional.**

O **CP/M** (Control Program for Microcomputers) é um sistema operacional produzido pela companhia americana DIGITAL RESEARCH, destinado a rodar, com algumas alterações, em qualquer microcomputador baseado em 8080, 8085 ou Z-80 (já existem versões para 8088/8086), que possua pelo menos 20 Kbytes de memória RAM e de uma até dezesseis unidades de disco.

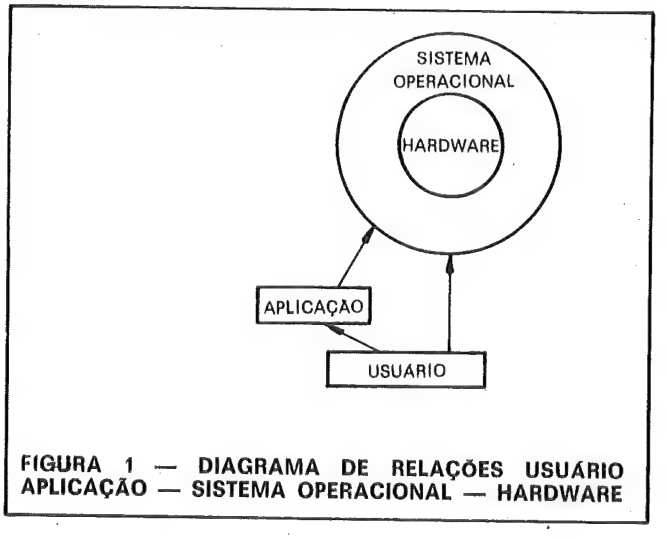


FIGURA 1 — DIAGRAMA DE RELAÇÕES USUÁRIO APLICAÇÃO — SISTEMA OPERACIONAL — HARDWARE

O CP/M proporciona um ambiente geral para construção, armazenamento, montagem ou compilação e execução de programas em assembler ou linguagens de alto nível.

HISTÓRICO

O CP/M foi desenvolvido em 1973 por Gary Kildall, na época, um consultor de software para a INTEL.

Por volta de 1975, um número significativo de companhias americanas estava fazendo microcomputadores. A maioria destas preferiu desenvolver os seus próprios sistemas operacionais. Tivessem estes pioneiros (ALTAIR, POLYMORPHIC, PROCESSOR TECHNOLOGY, etc.) sido capazes de fazer seus produtos chegarem rapidamente aos consumidores, o CP/M talvez não fosse o sistema operacional quase padrão que é hoje.

Ao invés disso, alguns pequenos fabricantes (TARBELL ELECTRONICS, DIGITAL MICROSYSTEMS, etc.) contornaram este custoso e demorado desenvolvimento, adotando o CP/M, e conseguiram por os primeiros sistemas com disco no mercado.

A chave para tornar o desenvolvimento de software financeiramente viável, era desenvolver programas que rodassem em diversos microcomputadores distintos. O CP/M tornou isto possível, desde que ele é capaz de rodar em qualquer microcomputador baseado em 8080, 8085 ou Z80 usando qualquer unidade de disco.

Existe hoje, rodando em CP/M, uma vasta gama de compiladores e interpretadores para as mais diversas linguagens; editores de texto; processadores de palavra; assemblers; gerenciadores de bancos de dados e os mais diversos programas de aplicação.

No decorrer do tempo, o CP/M sofreu algumas modificações, tendo sido emitidas quatro ou cinco versões. A versão corrente é a 2.2, na qual basearemos nosso trabalho.

O CP/M standard é fornecido em disco flexível de 8", formato IBM 3740 (densidade simples, 77 trilhas, 26 setores por trilha). Existem no mercado várias versões em diversos formatos, tanto em 8" como em 5 1/4".

A ESTRUTURA

O CP/M é normalmente carregado no topo da memória disponível e divide-se logicamente em quatro partes distintas:

BIOS – Basic Input/output System (o único módulo dependente do hardware)

BDOS – Basic Disk Operating System

CCP – Console Comand Processor

TPA – Transient Program Area.

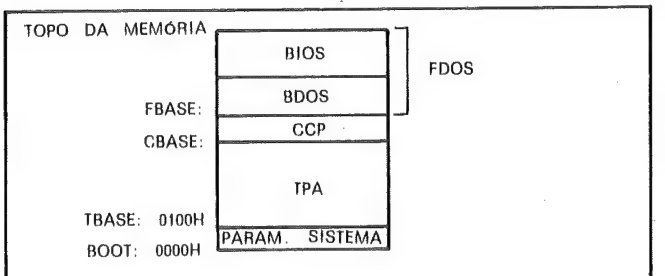


FIGURA 2 — MAPA DE MEMÓRIA DO CP/M

O **BIOS** contém as rotinas básicas de entrada e saída necessárias para acessar as unidades de disco ao teclado, ao vídeo, à imprensora etc. O acesso ao BIOS é feito através de uma tabela de jumps, localizada no início do mesmo, que apontam para as subrotinas internas. O BDOS (único módulo a chamar o BIOS) localiza as rotinas no BIOS chamando a posição da tabela de jumps correspondente à rotina desejada.

O CP/M pode ser adaptado a qualquer hardware particular pela alteração deste módulo. O BIOS, é normalmente escrito pelo fabricante do equipamento e posto no lugar do BIOS original, passando a chamar-se **CBIOS** (Custom BIOS). O CP/M standard é fornecido com um BIOS para o MDS 800 da INTEL.

Além das primitivas de entrada e saída, o BIOS contém o bloco de parâmetros de disco (DPH/DPB) que define as características de cada disco. Este bloco contém, entre outras, informações sobre: número de setores por trilha, tamanho do bloco em setores, tamanho do disco em blocos, número de entradas no diretório, número de trilhas reservadas para o sistema, etc.

O BIOS contém ainda, duas rotinas de inicialização (BOOT) e recarga (WBOOT) utilizadas quando o sistema é ligado, resetado ou reinicializado por programa.

O BDOS, o núcleo propriamente dito do CP/M, é responsável pelo gerenciamento das operações de disco, console e impressora. O BDOS controla até dezesseis unidades de disco, possibilitando a manutenção de arquivos seqüenciais ou randômicos de até 8 Mbytes cada. É ele quem cuida da alocação de espaço em disco; criação, abertura e fechamento de arquivos; escrita e leitura de arquivos; pesquisa de arquivos no diretório; seleção de discos; troca de nome e apagamento de arquivos, etc. Para o BDOS, os arquivos são mantidos em registros de tamanho fixo de 128 bytes.

"O CP/M sofreu algumas modificações, tendo sido emitidas quatro ou cinco versões".

O BDOS executa trinta e oito funções diferentes. Essas funções são chamadas através de um **CALL** para o ponto de entrada no endereço 0005H. Ao ser executado este CALL, o registrador C deve conter o número da função desejada, e o par DE o parâmetro ou o endereço do parâmetro, conforme a função a ser executada.

Entre outras, o BDOS executa as seguintes funções:

- seleciona um drive;
- cria um arquivo;
- abre um arquivo;
- fecha um arquivo;
- busca arquivos no diretório;
- deleta um arquivo;
- muda o nome de um arquivo;
- lê ou escreve em um arquivo, seqüencialmente;
- lê ou escreve em um arquivo, randomicamente;
- modifica os atributos de um arquivo.

O CCP proporciona a interface entre o usuário e o restante do sistema operacional. Ele é o responsável pela leitura, interpretação e execução dos comandos teclados pelo usuário.

O CCP pode acessar a todos os discos logicamente ligados ao sistema (disco pode estar fisicamente ativo, mas logicamente inativo). O CCP utiliza as letras de A até P, para indicar qual a unidade de disco que está sendo usada. Após a inicialização do sistema, o CCP assume a unidade "A", mostrando no vídeo: A

Isto indica que o CCP está pronto para receber comandos, e que se nesses comandos não for especificada a unidade, o sistema assumirá "A".

O CCP possui sete comandos residentes:

d	– seleciona a unidade de disco "d"
DIR	– lista o diretório de um disco
ERA	– apaga um arquivo
TYPE	– lista o conteúdo de um arquivo
REN	– troca o nome de um arquivo
SAVE	– salva o conteúdo da memória em disco
USER	– troca o número do usuário atual.

Se o comando emitido pelo usuário não coincide com nenhum dos sete acima mencionados, o CCP procura no diretório do disco especificado um programa do tipo COM (vide referência e arquivos), e se o encontra, carrega-o no início da área de transientes (TPA), no endereço 100H, e passa a executá-lo. Os programas assim carregados são normalmente chamados de **comandos transientes**. Após a execução, o transiente devolve o controle ao CCP, que passa a aguardar novos comandos.

O CP/M é fornecido com alguns comandos transientes:

START	– verifica o espaço ocupado por um arquivo, o espaço disponível em disco, as características de um disco, etc.
PIP	– copia arquivos e movimenta arquivos entre periféricos
DUMP	– mostra o conteúdo de um arquivo byte a byte em hexadecimal
ED	– editor de textos
DDT	– debugger
MOVCPM	– adapta o CP/M a qualquer tamanho de memória, entre 20 e 64 Kbytes
SYSGEN	– gera uma cópia do CP/M em diskette
ASM	– assembler para 8080
LOAD	– converte o código HEX (formato INTEL) gerado pelo ASM em um programa objeto (tipo COM) pronto para ser executado como transiente.

O usuário pode desenvolver seus próprios programas transientes, utilizando compiladores para linguagem de alto nível ou assemble (ASM) fornecido com o CP/M.

REFERÊNCIA A ARQUIVOS

No CCP os arquivos são referenciados por um identificador composto de duas partes: o nome e o tipo. O nome e o tipo são separados por um ponto. como no exemplo abaixo:

nnnnnnnn.ttt

Neste exemplo, nnnnnnnn representa o nome de até oito caracteres ASCII, e ttt, o tipo opcional, com até três caracteres. Se o tipo não é especificado, o sistema assume três brancos (20H). O tipo normalmente é utilizado para especificar se o arquivo é um programa escrito em assembler, BASIC, PL1, ou um texto, um cadastro de pessoal, etc. O nome serve para distinguir um arquivo em particular. Por exemplo:

LISTA. ASM é o código fonte do programa LISTA, enquanto:

LISTA, OBJ é o código objeto do mesmo programa.

A referência a um arquivo pode ser ambígua ou direta. Uma referência direta é aquela que identifica unicamente um arquivo, enquanto a referência ambígua pode ser satisfeita por um número de arquivos diferentes.

Uma referência ambígua é normalmente utilizada para pesquisar em diretório ou deleção de arquivos. A referência ambígua é similar à referência direta, a não ser pelo fato de que o símbolo "?" pode ser inserido em qualquer posição, tanto no nome como no tipo, indicando que nesta posição qualquer caracter vale. Assim, a referência ambígua: **T?STE.BAS** é satisfeita por qualquer um dos seguintes identificadores:

TEST.BAS
TOSTE.BAS
TISTE.BAS.

Um * no campo de nome ou tipo, equivale a preencher este campo com "?". Assim sendo, a referência:

* * *

Equivale à:

????????.???

Enquanto:

*ASM

E

CARTA.*

São satisfeitas por:

CARTA.ASM

Uma referência pode ainda conter antes do nome, uma letra (de A a P) seguida de ":", a fim de identificar o disco em que se encontra o arquivo.

A referência: B:JORNAL.TEX indica que o arquivo JORNAL.TEX está no disco B.

A TPA, que inicia no endereço 01000H conforme mencionado anteriormente, é a área onde são carregados para execução, tanto os comandos transientes do sistema como os programas de aplicação.

O CCP, ao passar o controle para o programa a ser executado, assume que o ponto de entrada do mesmo é 0100H e executa um CALL para este endereço.

Se o ponto de entrada do programa não é 0100H, esta posição deve conter um jump para o ponto de entrada real. Alguns assemblers já fazem isto automaticamente, ao montar o programa objeto.

A área de 0000H até 0100H contém uma série de parâmetros utilizados pelo sistema e pelos transientes, tais como: ponto de entrada da rotina de reinicialização, BOOT (0000H); o ponto de entrada do BDOS (0005H); o número do disco atual para o CCP (0004H); os vetores de interrupção, se forem utilizados; um buffer para operações de disco, etc.

POSICIONAMENTO EM DISCO E PROCESSO DE CARGA

O CP/M (CCP + BDOS + BIOS) se encontra residente nas duas primeiras trilhas de cada disco e é carregado na memória RAM cada vez que o sistema é ligado, resetado ou reinicializado. A carga é sempre feita a partir do disco "A".

A carga inicial, na maioria dos casos, é efetuada da seguinte forma: ao ser ligado, o sistema executa um pequeno programa, residente em ROM, que traz para a memória o primeiro setor da trilha 00 do drive "A", o qual contém um programa chamado **cold start loader**, que carrega todo o CP/M e passa o controle para a rotina BOOT no BIOS. Em alguns sistemas mais complexos, a carga do **cold start loader** é feita, automaticamente, pelo controlador de discos.

A tabela abaixo mostra a distribuição do CP/M nas trilhas 0 e 1 de um disco padrão IBM 3740.

TRILHA 00	SETOR 01	END. DE CARGA (VARIA C/O SIST).	MÓDULO DO CP/M COLD START LOADER
"	02	3400H+b	CCP
"	03	3480H+b	"
"	04	3500H+b	"
"	05	3580H+b	"
"	06	3600H+b	"
"	07	3680H+b	"
"	08	3700H+b	"
"	09	3780H+b	"
"	10	3800H+b	"
"	11	3880H+b	"
"	12	3900H+b	"
"	13	3980H+b	"
"	14	3A00H+b	"
"	15	3A80H+b	"
00	16	3B00H+b	CCP
"	17	3B80H+b	"
00	18	3C00H+b	BDOS
"	19	3C80H+b	"
"	20	3D00H+b	"
"	21	3D80H+b	"
"	22	3E00H+b	"
"	23	3E80H+b	"
"	24	3F00H+b	"
"	25	3F80H+b	"
"	26	4000H+b	"
01	01	4080H+b	"
"	02	4100H+b	"
"	03	4180H+b	"
"	04	4200H+b	"
"	05	4280H+b	"
"	06	4300H+b	"
"	07	4380H+b	"
"	08	4400H+b	"
"	09	4480H+b	"
"	10	4500H+b	"
"	11	4580H+b	"
"	12	4600H+b	"
"	13	4680H+b	"
"	14	4700H+b	"
"	15	4780H+b	"
"	16	4800H+b	"
"	17	4880H+b	"
"	18	4900H+b	"
"	19	4980H+b	"
"	20	4A00H+b	BIOS
"	21	4A80H+b	"
"	22	4B00H+b	"
"	23	4B80H+b	"
"	24	4C00H+b	"
"	25	4C80H+b	"
"	26	4D00H+b	"

As trilhas de 2 a 76 são utilizadas para armazenar o diretório e os arquivos.

Em virtude do CP/M ser relocável em função do tamanho da memória disponível (através do comando MOVCPM), a coluna **endereço de carga** da tabela, mostra o endereço de carga de cada setor com um valor básico mais um **bias**. O valor básico é o endereço de carga para o CP/M em 20 Kbytes, que é o menor tamanho de memória na qual o CP/M 2.2 pode rodar em condições normais. O **bias** pode ser calculado para qualquer tamanho de memória entre 20 e 64 Kbytes, da seguinte maneira:

b = tm x 1024 – 20480

onde: tm = tamanho de memória

PÁGINA O

A área de memória entre 0000H e 0100H contém diversos parâmetros que são utilizados durante a execução do CP/M. Os endereços e funções de cada parâmetro são apresentados a seguir:

Endereço:	conteúdo/função:
0000H a 0002H	Contém uma instrução de JUMP para a rotina de partida a quente (WB00T) no BIOS em 4A03H + b. Este JUMP é executado toda vez que o sistema é resetado ou algum programa, inclusive o CP/M, salta para o endereço 0000H.
0003H	IOBYTE (opcional) descrito mais adiante.
0004H	Disco corrente para o CCP (0 = A, 15 = p) Se em um comando o operador não especifica o drive, o CCP assume o valor contido nesta posição.
0005H a 0007H	Contém uma instrução de JUMP para o ponto de entrada do BDOS. Este JUMP é normalmente utilizado de duas maneiras: um CALL 0005H possibilita o acesso às funções executadas pelo BDOS e um LHLD 0006H põe em HL o endereço do BDOS. Esta segunda modalidade permite que um programa "saiba" a última posição de memória disponível da área de programas transientes. TPA. Esta facilidade é usada pelo DDT, que após carregado em 0100H se auto-reloca para a área de memória 4 Kbytes abaixo do BDOS (sobrepondo-se ao CCP) e altera o operando do JUMP em 0005H para 4K a menos. Isto possibilita ao DDT carregar programas para depuração no endereço normal de carga (0100H).
0008H a 0037H	Não utilizado pelo CP/M.
0038H a 003AH	**Restart 7** – Quando o DDT está operando como **debugger**, esta posição contém um JUMP para o interior do mesmo. A instrução RST 7 é utilizada como **breakpoint** pelo DDT.
0038H a 0058H	Não utilizado pelo CP/M.
005CH a 007FH	Nesta área o CCP e alguns programas transientes montam um FCB (File Control Block). A estrutura do FCB será assunto da próxima parte deste artigo.

O BIOS

Conforme mencionado anteriormente, o BIOS contém todas as rotinas básicas de E/S e possui, como ponto de entrada, uma tabela de JUMPS que apontam para as rotinas dentro do mesmo. Pode-se ver abaixo a distribuição e os endereços de cada JUMP.

```
4A00H + b JMP BOOT:   Inicializa o sistema
4A03H + b JMP WBOOT:  Partida a quente
4A06H + b JMP CONST:  Verifica estado do console
4A09H + b JMP CONIN:  Lê um caracter do console
4A0CH + b JMP CONOUT: Escreve um caracter no
                      console
4A0FH + b JMP LIST:   Escreve um caracter na
                      impressora
4A12H + b JMP PUNCH:  Escreve um caracter no
                      perfurador de fita
4A15H + b JMP READER: Lê um caracter leitor de fita
4A18H + b JMP HOME:   Cabeça do drive p/ trilho 0
4A1BH + b JMP SELDSK: Seleciona disco
4A1EH + b JMP SETTRK: Seta número da trilha
4A22H + b JMP SETSEC: Seta número do setor
4A25H + b JMP SETDMA: Seta endereço para buffer
                      de disco
4A28H + b JMP READ:   Lê o setor selecionado
4A2BH + b JMP WRITE:  Escreve o setor selecionado
4A2DH + b JMP LISTST: Verifica estado da impres-
                      sora
4A30H + b JMP SECTRAN:Translada setores lógicos
                      em setores físicos
```

Estas rotinas podem ser divididas em dois grupos: o das rotinas de tratamento de periféricos que manipulam caracteres, e o das rotinas de disco.

O CP/M "vê" quatro dispositivos **lógicos** de manipulação de caracteres, aos quais podem ser associados diversos dispositivos **físicos**. Estes dispositivos **lógicos** são:

CONSOLE: O dispositivo através do qual o operador interage com o sistema. Normalmente um terminal de vídeo ou um conjunto vídeo-teclado residentes.

LIST: Dispositivo para emissão de listagens (hard copy), normalmente uma impressora.

PUNCH: Este dispositivo correspondia nos primeiros sistemas CP/M, ao perfurador de fita de papel, equipamento bastante utilizado na época em que o CP/M foi desenvolvido. Atualmente, este dispositivo lógico é associado, normalmente, a um dispositivo físico do tipo modem, unidade de fita magnética etc.

READER: Idem ao PUNCH para a leitora de fita de papel.

O IOBYTE

Para possibilitar a associação de diversos dispositivos físicos a um dispositivo lógico alternadamente, o CP/M suporta a implementação do conceito de IOBYTE.

IOBYTE é um byte reservado de memória (0003H) que identifica qual o dispositivo físico que está associado a cada dispositivo lógico. O IOBYTE é tratado como quatro indicadores de dois bits, de acordo com o diagrama a seguir:

BIT	7	6	5	4	3	2	1	0
DISPOTISIVO	LIST		PUNCH		READER		CONSOLE	

Cada indicador pode assumir um valor entre 0 e 3, permitindo associar até 4 dispositivos físicos a cada dispositivo lógico, conforme a seguinte tabela:

DISPOSITIVO LÓGICO	DISPOSITIVO FÍSICO			
	00	01	10	11
CONSOLE:	TTY	CRT	BAT	UC1
READER:	TTY	RDR	UR1	UR2
PUNCH	TTY	PUN	UP1	UP2
LIST:	TTY	CRT	LPT	UL1

Onde:

TTY = impressora com teclado tipo teletype

CRT = terminal de vídeo ou conjunto vídeo – teclado residente.

LPT = impressora (Line-Printer).

UL1 = impressora auxiliar.

ADR = leitor de fita de papel.

PUN = perfurador de fita de papel.

BAT = modo **BATCH** no qual o dispositivo READER é utilizado como entrada de CONSOLE e o dispositivo LIST como saída.

UC1 = console auxiliar.

UR1 = leitor de fita auxiliar 1.

UR2 = leitor de fita auxiliar 2.

UP1 = perfurador de fita auxiliar 1.

UP2 = perfurador de fita auxiliar 2.

O BIOS contém ainda o bloco de parâmetros de disco, que será apresentado detalhadamente na terceira parte deste artigo, juntamente com o processo de alocação em disco.

Como o IOBYTE geralmente não é implementado, as rotinas CONST, CONIN, CONOUT e LIST serão descritas para um sistema com a seguinte configuração:

```
CONSOLE (ENTRADA) = TECLADO
        (SAÍDA)   = VÍDEO
LIST              = IMPRESSORA SERIAL
                    COMUM
```

AS ROTINAS DO BIOS

BOOT: Esta rotina é responsável pela inicialização do sistema durante a partida a frio. O carregador de partida a frio **(cold start loader)** passa o controle para BOOT logo após a carga do CP/M. BOOT efetua a inicialização de periféricos (se necessária), envia a mensagem inicial para o console e prepara contadores e parâmetros diversos, inclusive todos aqueles mencionados a seguir, na descrição de WBOOT.

WBOOT: Esta rotina é responsável pela partida a quente (warm start) do sistema. Isto ocorre quando o operador pressiona o botão de RESET, tecla controle ou quando um programa transiente executa um JUMP para o endereço

0000H, o qual contém uma instrução de JUMP para WBOOT.

É responsabilidade de WBOOT recarregar a partir do disco "A" todo o CP/M, com exceção do BIOS. Feito isso, WBOOT tem de inicializar os seguintes parâmetros de sistema:

ENDEREÇOS:	CONTEÚDO:
0,1,2	JUMP para WBOOT (0000H: JMP 4A03H + b)
3	Valor inicial para IOBYTE (se implementado)
5,6,7	JUMP para o BDOS (0005H: JMP 3C06H + b)

Após a inicialização, WBOOT executa um JUMP para o CCP (3400H + b) passando no registrador C o número do disco que o CCP deve assumir. Na maioria das implementações de BIOS, o valor passado em C é igual ao valor contido no endereço 0004H, onde o CCP mantém o número do disco ativo **(current disk)**. Isto faz com que o CCP volte a assumir o disco que estava utilizando antes da reinicialização.

CONST: Verifica o estado do teclado. Se houver tecla pressionada ou pendente, retorna com o registrador A = = FFH; do contrário, retorna com A = 00.

CONIN: Espera que uma tecla seja pressionada e retorna com o código ASCII correspondente no registrador A. Esta rotina deve eliminar o bit de paridade (bit 7 = 0) caso ele exista.

CONOUT: Envia o carácter (ASCII) recebido no registrador C para o vídeo.

LIST: Envia o carácter (ASCII) recebido no registrador C para a impressora.

PUNCH: Envia o carácter (ASCII) recebido no registrador C para o dispositivo físico associado ao dispositivo lógico **PUNCH**. Esta rotina, quando implementada, é geralmente utilizada para tratar a saída para modem, unidade de fita magnética etc.

READER: Lê um caractere do dispositivo físico associado ao dispositivo lógico **READER** e retorna com o código ASCII correspondente no registrador A. Esta rotina, quando implementada, é geralmente utilizada para tratar a entrada para modem, unidade de fita magnética etc.

HOME: Posiciona a cabeça de escrita e leitura do drive selecionado na trilha 00. Esta rotina é geralmente implementada usando-se o comando **RESTORE** existente em quase todos os contoladores de disco (1771, 1791, 8272, etc). HOME pode ainda ser implementada chamando-se SETTRK com BC = 0000.

SELDSK: Seleciona o drive especificado pelo registrador C para as próximas operações de disco. O valor passado em C varia de 0 para o drive A, a 15 para o drive P. Na maioria dos microcomputadores esta rotina só aceita valores entre 0 e 3 (4 drives). Ao retornar, SELDSK deve devolver em HL o endereço de uma tabela de parâmetros de disco chamada: **Disk Parameter Header**, que será descrita detalhadamente na próxima parte deste artigo. Caso o drive a ser selecionado não exista ou esteja inoperante, SELDSK deve retornar HL = 0000H para indicar erro. Neste caso, o BDOS mostra a mensagem:

"BDOS Err On X: Select"

SETTRK: Posiciona a cabeça de escrita e leitura do drive correntemente selecionado na trilha indicada pelo par BC (0 a 76 para o CP/M standard e de 0 a 65535 para as demais implementações). SETTRK não tem, obrigatoriamente, que mover a cabeça, podendo apenas atualizar um indicador de trilha, deixando a tarefa de posicionar a cabeça para as rotinas READ e WRITE.

SETSEC: Informa através do par BC o número do próximo setor a escrever ou ler. SETSEC pode passar o número do setor diretamente ao controlador de discos, ou salvá-lo na memória, deixando esta função para as rotinas READ e WRITE.

SETDMA: Esta rotina recebe em BC o endereço inicial do buffer (128 bytes) a ser utilizado nas próximas operações de escrita e leitura de setor. Por exemplo: se BC = 0080H quando SETDMA for chamada, então todas as operações subseqüentes de escrita e leitura irão carregar/ler dados de/para a área de memória entre 0080H e 100H.

READ: Lê o setor especificado através de SETSEC para o **buffer** especificado por SETDMA. Ao retornar, READ deve passar 00H no registrador A se não houve erro durante a leitura ou, caso contrário, A = 01H READ deve tentar a leitura durante pelo menos dez vezes, antes de retornar a indicação de erro, quando o BDOS mostra a mensagem:

"Bdos Err On X: Bad sector"

Se deseja-se um tratamento de erro mais apurado, este deve ser implementado no BIOS, antes de retornar indicando o erro. Por exemplo: pode-se informar ao operador o tipo de erro ocorrido como: CRC, busca, registro não encontrado, perda de danos, disco inoperante etc.

WRITE: Escreve o conteúdo do **buffer** especificado por SETDMA, no setor indicado por SETSEC. O tratamento de erro é idêntico ao mencionado para READ. Para manter compatibilidade com outros sistemas que utilizem CP/M, os setores devem ser gravados com o **data address mark**; igual a **No deleted data**.

LISTST: Verifica o estado da impressora. Se a mesma estiver em condições de receber um caracter (ASCII), retorna com o registrador A = FFH; caso contrário A = 00H. Esta rotina não é chamada pelo BDOS. Ela é utilizada, geralmente, por **SPOOLERS** para evitar que o console fique "preso" durante a espera pela impressora.

SECTRAN: Executa a translação de setores lógicos para setores físicos SECTRAN recebe em BC o número do setor lógico e em DE o endereço da tabela de translação de setores (este endereço é um dos parâmetros residentes nos **Disk Parameter Header** mencionados na descrição de SELDSK). O número do setor, passado em BC, é utilizado como indexador na tabela de translação para obter o número do setor físico, devolvido em HL.

A translação de setores é efetuada a fim de possibilitar o uso da técnica de **entrelaçamento de setores**. Esta técnica consiste em gravar os setores lógicos, não em setores fisicamente adjacentes, mas em setores físicos separados por um determinado número de setores, chamado **fator de entrelaçamento**.

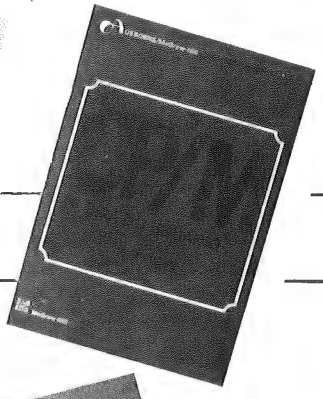
O uso deste recurso se justifica se imaginarmos o seguinte caso: é necessário efetuar a leitura de dois setores logicamente consecutivos (caso comum na carga de programa e leitura de arquivos seqüenciais). Se eles estiverem gravados em setores fisicamente adjacentes ocorrerá o seguinte: após a leitura do 1º. setor, transferência dos dados para área de memória onde devem ser carregados, e eventual processamento em cima deles, a cabeça de leitura e gravação do disco já terá passado o 2º. setor e terá de esperar o disco

O que distingue HELPWARE de outros livros de computação? Em primeiro lugar, a precisão. Se não é preciso, não é HELPWARE. Uma escrita clara e concisa, cuidadosamente editada, e o nosso compromisso de publicar somente aqueles livros que o ajudarão a obter um trabalho rápido, fácil e eficiente.

Antes de investir seu dinheiro num sistema de computação, invista em WELPWARE. Leia o resumo dos temas de seu interesse e preencha o coupon para efetuar sua compra.

HELPWARE

LIVRO EM PORTUGUÊS



CONSTRUA O SEU PRÓPRIO COMPUTADOR USANDO O MP-Z80 CIARCIA, Steve
Cr$ 21,900,

Este guia prático mostra como construir um computador, baseado no famoso microprocessador, o Zilog Z80. A descrição enfoca um microcomputador básico em placa única, contendo 2K de sistema operacional, portas serial e paralelas, display hexadecimal e armazenagem de massa em fita cassete. Cada subsistema do computador é completamente explicado e calçado em informações provadas e testadas, de forma que o leitor possa facilmente modificar o sistema — adicionando por exemplo um terminal de vídeo — a fim de satisfazer suas necessidades pessoais.

INICIAÇÃO AOBASIC FOX, Annie e FOX, David
CR$ 9.000,

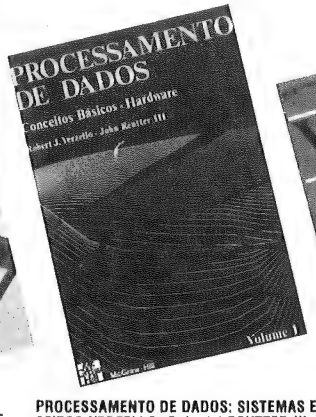
Este é um livro especial de introdução a mais popular de todas as linguagens de programação: Basic. Os autores escreveram INICIAÇÃO AO BASIC especialmente para iniciantes que não tenham acesso aos micros, mas que gostariam de familiarizar-se com conceitos de programação.

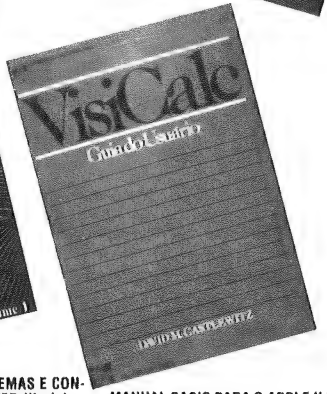
PROGRAMAÇÃO COM BASIC GOTTFRIED, Byron
CR$ 11.200,

Este livro tem por finalidade fornecer um curso de programação de computadores, empregando a estrutura padronizada da linguagem Basic. Assim sendo, todas as características principais da linguagem serão analisadas, fornecendo ao leitor condições de organizar e escrever eficientes programas de computadores. A obra foi organizada em duas partes (1) Basic Básico e (2) Basic Avançado.

CP/M GUIA DO USUÁRIO HOGAN, Thom
CR$ 13.900,

CP/M (Control Monitor Program) tem se tornado o sistema operacional padrão para os microscomputadores baseados no 8080 e Z80. Nesta edição do CP/M, o autor expandiu o texto para acompanhar os mais recentes desenvolvimentos, incluindo o CP/M-86, sistema operacional baseado no 8086 e no 8088 para micros, tais como o computador pessoal da IBM. Escrito para usuários em todos os níveis de conhecimento especializado, este livro considera a história e funções do CP/M, bem como os comandos próprios para o usuário.





APPLE II — GUIA DO USUÁRIO POOLE, Lon / McNIFF, Martin / COOK, Steve
CR$ 22.900,

Este livro informa o leitor sobre o Apple, mais do que qualquer outra fonte. Ele cobre as programações Applesoft e o Integer Basic para mostrar como fazer o melhor com com o Apple, cores e capacidades gráficas. Contém uma rigorosa descrição de toda instrução em Basic, comandos e funções e fornece também uma secção especial em programação avançada e aplicações. Apple II descreve características de Hardware, tais como drive de disquetes e impressoras. Escrito em linguagem clara e rigorosamente organizado, este é o guia verdadeiramente compreensivo que todos devem ter perto de seus terminais.

Introdução aos microcomputadores, OSBORNE, Adam e BUNNEL, David Vol. 0
CR$ 10.400,

Esta terceira edição, revista e ampliada, do volume 0 transmite ao iniciante uma extensa cobertura sobre microcomputadores: como eles funcionam, o que eles podem fazer pelos problemas de gerenciamento de dados nos dias atuais. O texto descreve claramente o componente de um sistema de microcomputador, introduzindo, ao leitor, linguagens de programação, códigos binários e aritméticos, lógica, temporização e memória. Novas secções em software e comunicações foram adicionadas.

PROCESSAMENTO DE DADOS: SISTEMAS E CONCEITOS VERZELLO, Robert / REUTTER III, John
2 VOLUMES CR$ 18.600
Cada CR$ 9.300,

Este texto básico é dirigido a cursos de Introdução a Processamento de Dados ou Introdução à Ciência dos computadores, ensinados em Administração e Economia. Os autores discutem como usar a tecnologia dos computadores para resolver problemas de qualquer tipo de linguagem ou tipo específico de máquina, eles discutem completamente todos os tópicos apropriados. O livro é ainda cuidadosamente projetado para estudantes com base limitada em matemática.

MANUAL BASIC PARA O APPLE II PECKHAM, Herbert
CR$ 11.700,

Este material é dirigido para cursos de programação Basic, oferecidos aos estudantes de ciências, que fazem o uso do computador pessoal Apple II. Possibilita ao leitor aprender como programar o Apple II, através de exercícios dirigidos.

LIVROS EM INGLÊS

THE APPLE PROGRAM FACTORY STEWART, George CR$ 44.900,

A collection of 25 fun-and-games programs for Apple home computers, adapted from his POPULAR COMPUTING column, "The Program Factory", by reporter George Stewart. Ideal for Apple users at all levels of ability, yt's a booc the entire family can enjoy. Some of the programs included can be used as aids in hobbies like weaving, cryptography and writing poetry. All are designed to offer hours of entertainment from your Apple home computer.

COMMUNICATION AND NETWORKING WITH THE IBM PC KRUGLINSK, David CR$ 55.000,

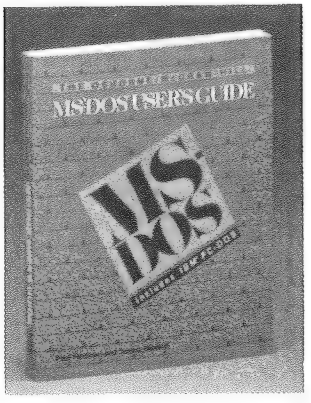
Find out how a serial communications network could enable your IBM PC to communicate with other IBM PCs, mainframe and mini computers. Combining theory, product descriptions and practical examples, David Kruglinsk highlights the fascinating communication techniques available now to large corporations, small businesses and the individual home user. No technicalskills are required to enjoy this exciting and informative look at the world af data communications and network.

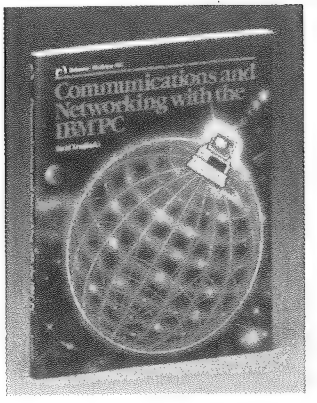
THE OSBORNE / McGRAW-HILL MS-DOS USER'S GUIDE HOFFMAN, Pau, CR$ 44.900,

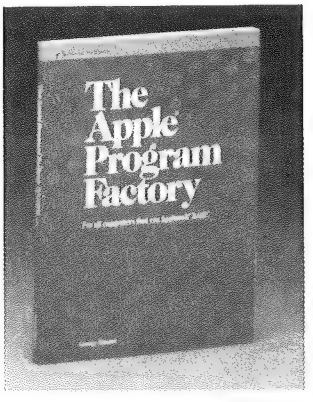
A comprehensive guide to the MS-DOS operating system, this book is designed to familiarize you with this powerful system from Microsoft in all its versions. Ideal for beginners and experienced users alike, this guide covers each computer running MS-DOS, gives the version it runs and any improvements the manufacturer has made to the system. It also gives complete information on the PC DOS version designed for the IBM PC. Additional programs and reference material make this guide a tool of lasting value.

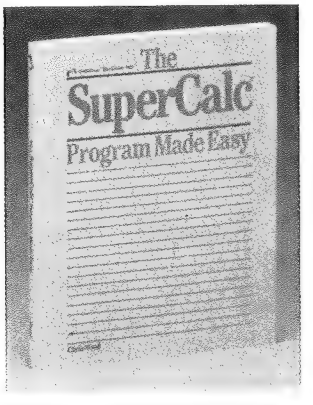
THE SUPERCALC PROGRAM MADE EASY WOOD, Chris CR$ 44.000,

Learn to produce effective budgets, reports, and business graphics with your SuperCalc software. This complete guide to SuperCalc (versions 1, 2, and 3) teaches you all the skills you need to take full advantage of this versatile program — from building and organizing a worksheet to using advanced problem-soving techniques and specail SuperCalc features. A handy reference section that lists and explains every SuperCalc command is also provided. With this book, even beginners will feel confident to construct and customize models that meet their own needs.



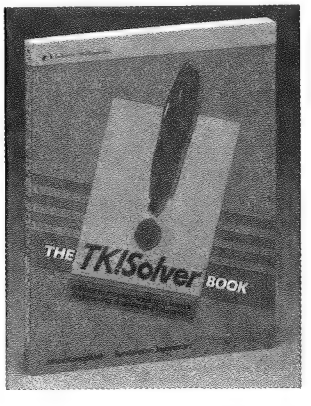






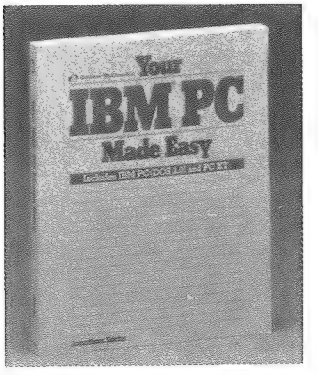
THE TK! SOLVER BOOK: A GUIDE TO PROBLEM-SOLVING IN SCIENCE, ENGINEERING, BUSINESS, AND EDUCATION KONOPASEK, Milos and JAYARAMAN, Sundaresan CR$ 69.000,

Now, from two of the creators of TK! Solver, a comprehensive guide designed to help you take advantage of this powerful, problem-solving software from Software Arts. Step-by-step, this book guides you through the world of TK! Solver, explaning its features and teaching you how to best utilize them. Concise case studies fully illustrate a broad range of TK! Solver applications in science, engineering, business, and education. Beginners nad computer professionals alike, can quickly master the subtleties of TK! Solver with the help of this simple, yet sophisticated, guide.



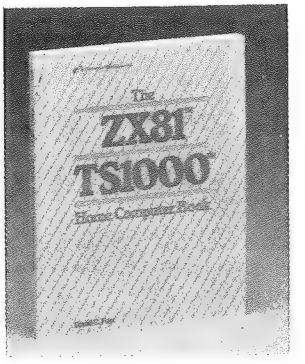
YOUR IBM PC MADE EASY (INCLUDES IBM PC (DOS 2.0) AND PC-XT) SACHS, Jonathan cr$ 44.000,

A basic operating guide for the beginning IBM PC Made Easy covers all the fundamentals of your new system. Detailed explanations of each major feature, step-by-step running instructions, software, use and care, tips on how to trouble shoot computer breakdowns — even a guide to IBM supplies, services, and user groups — it's all here in this easy-to-understand tutorial.



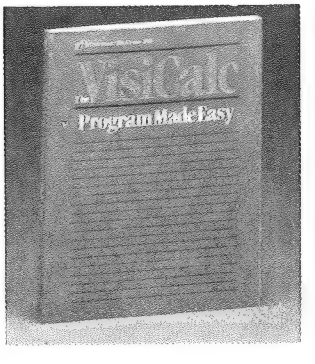
THE ZX81 / TS1000 HOME COMPUTER BOOK FOYT,David C. CR$ 41.000,

Learn how to operate these revolutionary low-cost computers, while developing valuable programing skills with The ZX81 / TS1000 Home Computer Book. After presenting simple instructions on how to operate the computers and their peripherals, the author introduces a systematic tutorial designed to teach you Basic. In addition to building programming skills, you'll learn about computer graphics, animation, and machine language techniques. The ZX81 / TS1000 Home Computer Book will also serve as a lasting reference tool because of its summary of BASIC commands, and memory map.



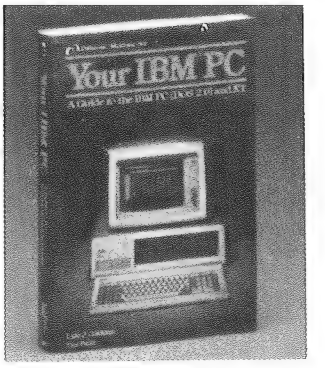
THE VISICALC PROGRAM MADE EASY CASTLEWITZ, David M. CR$ 44.000,

The exercises presented in The VisiClac Program Made Easy will teach you all the basic skills you need to design a professional worksheet and all the advanced skills required to tap the special capabilities of VisiCalc. You'll quickly maximize the quality of your output by reading this fundamental, computer-side tutorial.



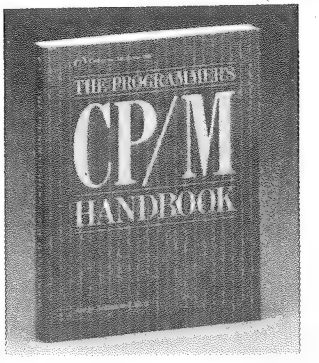
YOUR IBM PC: A GUIDE TO THE IBM PC (DOS 2.0) AND XT GRAHAM, Lyle and FIELD, Tim CR$ 63.000,

This popular, comprehensive user guide has been updated to include both the IBM PC (DOS 2.0) and the PC XT!



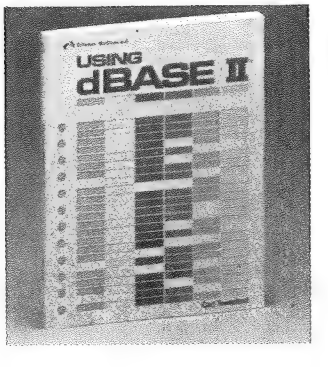
THE PROGRAMMER'S CP/M HANDBOOK JOHNSON-LAIRD,Andy CR$ 74.000,

An exhaustive coverage of CP/M-80, its internal structure and major components is presented in The Programmer's CP/M Handbook. Written for the programmer, this 750-page volume includes subroutine examples for each of the CP/M system calls and information on how to customize CP/M complete with detailed source codes for all examples. A dozen utility programs are shown with heavily annotated C-language source codes. An inviaiuable and comprehensive tool for the serious programmer.



USING dBASE II TOWNSEND, Carl CR$ 64.000,

Here's a manual for business professionals who want to design customizes programs with dBase II software. Informative, clear, and nontechnical in style, Using dBase II is a compendium of techniques presented to help you master this remarkable software package. After learning about installation, you'll be introduced to information file organization, records, and dabases. Then you'll move on to system design, structured programming, and other advanced topics. Using dBase II provides you with a virtual encyclopedia of information, much of wich is unavailable elsewhere.



WORDSTAR MADE EASY (Second Edition) ETTLIN, W.ALTER A. CR$ 44.900,

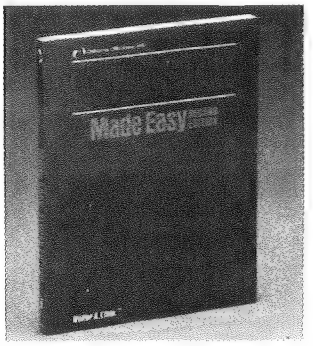
In just a few easy, time-saving lessons, you'll learn to produce sales reports, legal documents, business letters, manuscripts and more.Also featured is a useful pull-out card for quick reference and computing ease. Simple to learn and handy to use, WordStar Made Easy is highly recommended.

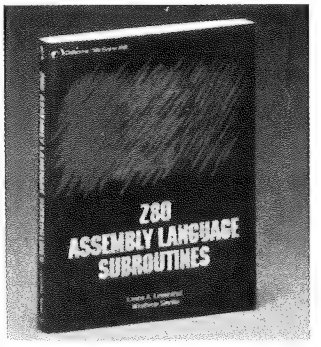
Z80 ASSEMBLY LANGUAGE SUBROUTINES LEVENTHAL, Lance A. and WINTHROP, Saville CR$ 65.000,

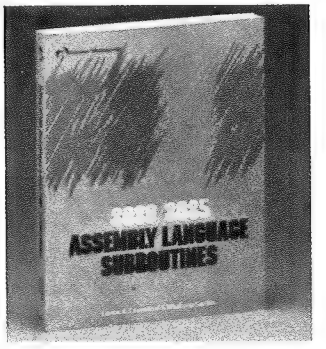
An overview of assembly language programing for the Z80, this book provides over 50 useful subroutines which will save you valuable programming time. It includes array, bit and string manipulation, code conversation, arithmetic operations, sorting and searching techniques, and much more. All routines have been throughly tested and debugged.

8080/8085 ASSEMBLY LANGUAGE SUBROUTINES LOVENTHAL, Lance A. and WINTHROP, Saville CR$ 59.500,

"The text is beautifully designed with many, many examples complete with source code. The routines included can actually be used to save hours of tedious development time. Whwther you program in assembly for fun or profit check this one out. You'll be impressed!
(Lifelines/The Software Magazine)







PARA ISTO ESTOU ENVIANDO CHEQUE NOMINAL À EDITORA INTERFACE

GOSTARIA DE ADQUIRIR OS SEGUINTES EXEMPLARES

NOME DO LIVRO

ADAVANCED

INTRODUCING THE UNIX SYSTEM McGILTON, H. and MORGAN, R. CR$ 70.000,

This right-to-pojnt introductory guide to the UNIX operating system will get you over the initial hurdles and hazards in using this powerful new computer tool. And for those with some UNIX experience, a better understanding of the system and its sometimes cryptic documentation. Thorough, detailed coverage, including shell programming, the ex-text editor, the vi display editor, text manipulation, the directory struture and file system, commands, and more.
A BAYTE book

SOFTWARE ENGINEERING
Software Defect Removal DUNN, R. H. CR$ 98.000,

This comprehensive volume thoroughly examines and analyses the full spectrum of techniques for detecting and removing "bugs" from computer programs during their development and subsequent modifications. Beginning with an overview of all defect removal methods, the author provides a comparison of the effectiveness of each method and how well it performs with respect to differing classes of defects it is designed to handle.

The McGraw Hill COMPUTER HANDBOOK HELMS, H. L. and Jr, Ed CR$ 245.000,

Here's the everything-between-two-covers reference tool the computer world has been waiting for! Written by a staff of experts, it answers virtually any question of either the experienced user or the beginner. Covers mainframes, minis, micros, gardware and software, languages, I/O, operating systems, databases, files and much more, including voice recognition, graphics plotters, and local networking.
Inexhaustible and indispensable.

DATA ADMINISTRATION DURELL, William R. CR$ 108.000,

A pratical guide for all individuals involved in the planning, development, or implementation of data administration, this unique book bypasses concept and theory and gets righ down to real-world information processing in the workplace. The effective integration of data administration is highly dependent upon the manner in wich it is "marketed", standardized, organized, and implemented within an organization. Logically processing through all these phase, the book present techniques for handling traditional DA problems through the not-so-traditional use of automated tools and rigorous standads and procedures. It delves into such concerns as securing the approval and commitment of upper management... promoting data sharing while reducing data redundancy... maintenance data integrity... increasing the flexibility and modularity of data... reducing future maintenance costs... and the placement of DA within the organization. Diagrams are used extensively to illustrate data structuring and decomposition. Data processing managers and administrators, as well as programmers and analysts, will find this guide to be an invaluable sourcebook of concrete solutions to the day-to-day problems often encountered in the world of data administration and information processing.

DISTRIBUTED DATABASES
Principles and Systems
CERI, Stefano — Research Associate
PELAGATTI, Giuseppe — Associate Professor
both of Politecnico di Milano
CR$ 85.000,

A distributed database is an integrated database for a computer network. Building and implementing distributed databases involves a range of new problems. In this book, readers will find an overall and coordinated presentation of the fundamental principles of this new technology. The book is intended for readers who have a backgroung in databases, but are not distributed database specialists.

dar mais uma volta completa para acessar este setor. Se escolhermos um entrelaçamento tal que o tempo de tratamento dos dados lidos corresponda ao tempo que a cabeça leva para passar pelos setores de separação definidos pelo entrelaçamento, quando solicitar-se a leitura do próximo setor a cabeça estará próxima do setor a ser lido. Com isso estarão otimizadas as operações de acesso ao disco, tanto de leitura como de gravação, exigindo menos voltas completas do disco para se transferir dados no sentido CPU = disco ou disco = CPU.

A figura abaixo mostra o diagrama de correspondência entre setores físicos e lógicos para um trilha de um disco padrão IBM 3740 utilizando um fator de entrelaçamento igual a 6.

Em alguns sistemas, o entrelaçamento já é efetuado na própria formatação do disco, tornando SECTRAN desnecessária.

SETOR FÍSICO	1	2	3	4	5	6	7	8	9	10	11	12	13
SETOR LÓGICO	0	13	9	22	5	18	1	14	10	23	6	19	2
SETOR FÍSICO	14	15	16	17	18	19	20	21	22	23	24	25	26
SETOR LÓGICO	15	11	24	7	20	3	16	12	25	8	21	4	17

O DIRETÓRIO

No CP/M o espaço de disco é alocado em blocos de igual tamanho de 1,2, 4,8 ou 16 Kytes. O tamanho do bloco varia de sistema para sistema e pode ser definido na tabela de parâmetros de disco no BIOS, descrita logo adiante. Na maioria dos microcomputadores que utilizam discos flexíveis (8" ou 5,1/4"), estes blocos são de 1 ou 2 Kbytes.

Na medida em que um dado arquivo vai crescendo, o BDOS vai alocando blocos para este e, quando o mesmo é apagado, todos esses blocos são liberados para uso por outros arquivos.

Internamente, todos os arquivos são logicamente divididos em extensões de 16 Kbytes a fim de facilitar a construção do diretório.

O CP/M mantém, geralmente a partir da trilha 2 de cada disco, um diretório de arquivos. Este diretório contém as informações relativas aos espaços alocados para todos os arquivos do disco e é composto por um determinado número de "entradas". Cada uma destas entradas consiste de uma seqüência de 32 bytes arranjados como mostra a figura 3.

Descreveremos agora, cada um dos campos desta entrada:

us — Nº do usuário em cuja sessão foi criado o arquivo.

f1 a f8 — Nome do arquivo em ASCII. Caso o nome do arquivo não tenha 8 caracteres, o restante do campo é preenchido com espaços (20H).

t1 a t3 — Tipo do arquivo em ASCII. Os bits 7 dos bytes t1 e t2 são utilizados como indicadores de atributos do arquivo, da seguinte forma:

- t1(7) = 1 ⇒ Arquivo só de leitura, não pode ser escrito.
- t2(7) = 1 ⇒ Arquivo de sistema, não é listado quando o usuário pede o diretório.

ex — Número da extensão desta entrada. A fim de manter a compatibilidade com a versão 1.4, este campo só assume valores de 0 a 31.

s1 — Não utilizado na versão 2.2. Sempre igual a 00.

s2 — Os bits de 0 a 3 deste campo são utilizados como continuação do campo ex, possibilitando endereçar até 512 extensões (arquivo de 8 Mbytes).

rc — Contador de registros (setores de 128 bytes) alocados para a extensão corrente. Este campo pode assumir valores de 0 a 127.

d0 a d15 — Estes bytes são utilizados para guardar o número dos blocos alocados para este arquivo. Quando o disco possui menos de 256 blocos, cada byte contém o número de um bloco. Do contrário, cada dois bytes contém o número de um bloco.

CAMPO	us	f1	f2	f3	f4	f5	f6	f7	f8	t1	t2	t3	ex	s1	s2	rc	d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15
BYTE	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

FIGURA 3 — COMPOSIÇÃO DE UMA ENTRADA DO DIRETÓRIO

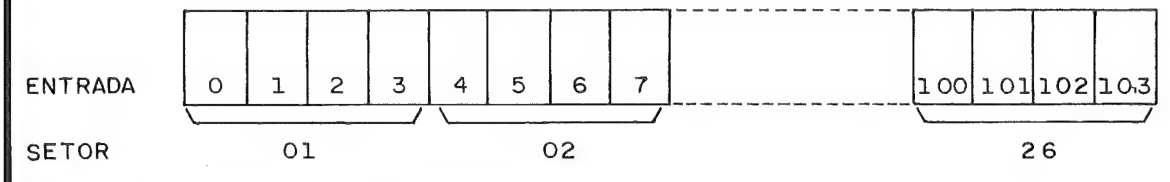


FIGURA 4 — DISTRIBUIÇÃO DAS ENTRADAS DE DIRETÓRIO EM UMA TRILHA DE UM DISCO FORMATO IBM 3740.

O FCB

Quando um programa deseja acessar a um determinado arquivo em disco, ele deve antes construir na memória um **File Control Block** (FCB) referente a este arquivo. O FCB consiste de uma seqüência de 33 bytes, quando o arquivo deve ser acessado seqüencialmente, ou 36 bytes, quando o acesso deve ser aleatório.

A formação de um FCB é quase idêntica a de uma entrada de diretório, e pode ser vista na figura 5.

Apresentamos abaixo a descrição dos campos que constitutem o FCB:

dr — Número do drive em que se encontra o arquivo.
dr = 0 — drive corrente
dr = 1 — drive A
dr = 2 — drive B
. . .
dr = 16 — drive P

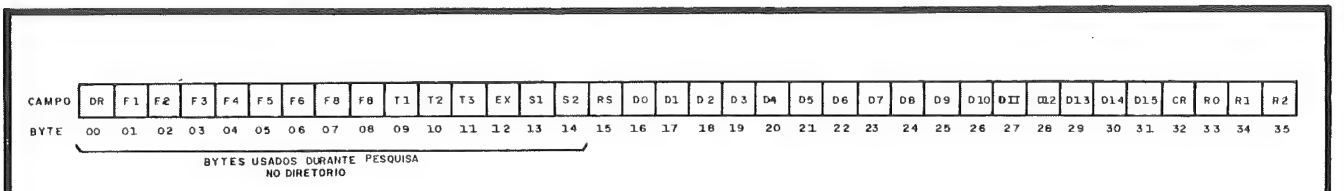


FIGURA 5 — COMPOSIÇÃO DO FCB

O CP/M possibilita a manutenção de arquivos de até 8 Megabytes (65536) registros de 128 bytes), os quais podem ser acessados de modo sequencial ou aleatório. Para que isso seja possível, o BDOS executa todo um processo de alocação de espaço em disco, procurando otimizar ao máximo o uso deste espaço e reduzir o movimento da cabeça de escrita e leitura e, consequentemente, o tempo de acesso aos dados desses arquivos.

f1 a f8 — Idem ao diretório.
t1 a t3 — Idem ao diretório.
ex — Número da extensão que se deseja acessar. Este campo é normalmente zerado pelo programa que tenta abrir o arquivo, o que possibilita o acesso ao primeiro registro do arquivo.
s1 — Idem ao diretório.
s2 — Bits 0 a 3 — Idem ao diretório.
bit 7 — Este bit é setado pelo BDOS toda vez que o arquivo é aberto. Se algum registro for escrito neste arquivo o bit em referência é **resetado**. Ao ser chamado para fechar o arquivo o BDOS verifica o estado deste bit. Se ele estiver **resetado**, o BDOS atualiza o diretório com as informações contidas nos bytes 12 a 31 do FCB e, caso ele permaneça **setado** o BDOS não atualiza o diretório, já que não foram alocados novos blocos para o arquivo. Isto é feito para minimizar o acesso ao diretório e, conseqüentemente o movimento da cabeça de escrita e leitura.
rc — Idem ao diretório.
d0 a d15 — Idem ao diretório.
cr — Número do registro a ser lido ou escrito na extensão corrente (ex). Este campo pode assumir valores de 0 a 127 e é automaticamente incrementado pelo BDOS após uma operação de escrita ou leitura.

CEDM

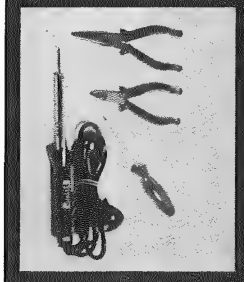
CURSOS DE APERFEIÇOAMENTO

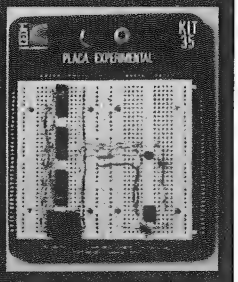
MAIS SUCESSO PARA VOCÊ!

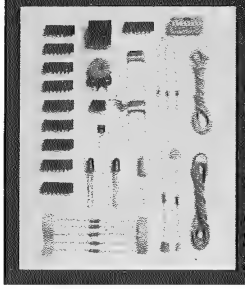
**Comece uma nova fase na sua vida profissional.
Os CURSOS CEDM levam até você o mais moderno ensino técnico programado e desenvolvido no País.**

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

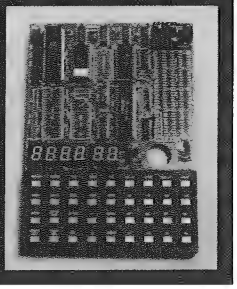
São mais de 140 apostilas com informações completas e sempre atualizadas. Tudo sobre os mais revolucionário CHIPS. E você recebe, além de uma sólida formação teórica, KITS elaborados para o seu desenvolvimento prático. Garanta agora o seu futuro.





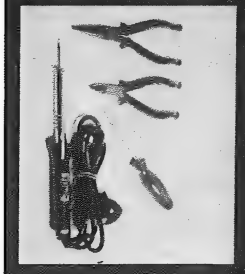
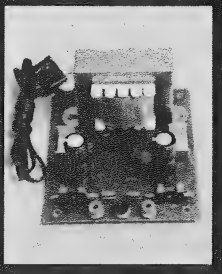
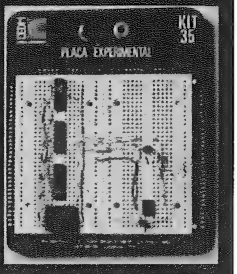


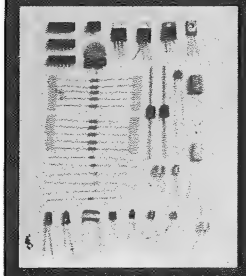
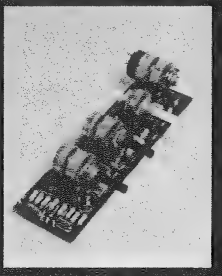
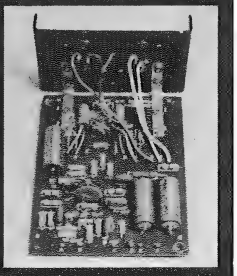
CEDM-20 - KIT de Ferramentas. CEDM-78 - KIT Fonte de Alimentação 5v/1A. CEDM-35 KIT Placa Experimental CEDM-74 - KIT de Componentes. CEDM-80 MICROCOMPUTADOR Z80 ASSEMBLER.



CURSO DE ELETRÔNICA E ÁUDIO

Métodos novos e inéditos de ensino garantem um aprendizado prático muito melhor. Em cada nova lição, apostilas ilustradas ensinam tudo sobre Amplificadores, Caixas Acústicas, Equalizadores, Toca-discos, Sintonizadores AM/FM, Gravadores e Toca-Fitas, Cápsulas e Fonocaptadores, Microfones, Sonorização, Instrumentação de Medidas em Áudio, Técnicas de Gravação e também de Reparação em Áudio.

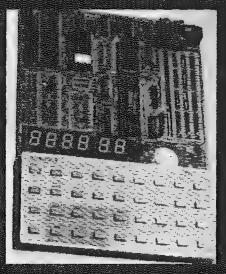
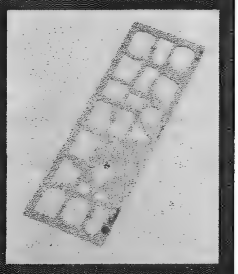




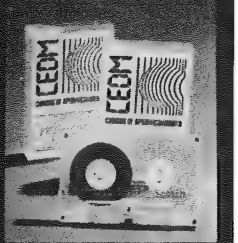
CEDM-1 - KIT de Ferramentas. CEDM-2 - KIT Fonte de Alimentação + 15-15/1A. CEDM-3 - KIT Placa Experimental CEDM-4 - KIT de Componentes. CEDM-5 - KIT Pré-amplificador Estéreo. CEDM-6 - KIT Amplificador Estéreo 40w.

CURSO DE PROGRAMAÇÃO EM BASIC

Este CURSO, especialmente programado, oferece os fundamentos de Linguagem de Programação que domina o universo dos microcomputadores. Dinâmico e abrangente, ensina desde o BASIC básico até o BASIC mais avançado, incluíndo noções básicas sobre Manipulação de Arquivos, Técnicas de Programação, Sistemas de Processamento de Dados, Teleprocessamento, Multiprogramação e Técnicas em Linguagem de Máquina, que proporcionam um grande conhecimento em toda a área de Processamento de Dados.

KIT CEDM Z80 BASIC Científico. KIT CEDM Z80 BASIC Simples. Gabarito de Fluxograma E-4. KIT CEDM SOFTWARE Fitas Cassete com Programas.



Você mesmo pode desenvolver um ritmo próprio de estudo. A linguagem simplificada dos CURSOS CEDM permite aprendizado fácil. E para esclarecer qualquer dúvida, o CEDM coloca à sua disposição uma equipe de professores sempre muito bem acessorada. Além disso, você recebe KITS preparados para os seus exercícios práticos.

Ágil, moderno e perfeitamente adequado à nossa realidade, os CURSOS CEDM por correspondência garantem condições ideais para o seu aperfeiçoamento profissional.

GRÁTIS

Você também pode ganhar um MICROCOMPUTADOR.

Telefone (0432) 23-9674 ou coloque hoje mesmo no Correio o cupom CEDM.

Em poucos dias você recebe nossos catálogos de apresentação.

CEDM

**Avenida São Paulo, 718 - Fone (0432) 23-9674.
CAIXA POSTAL 1642 - CEP 86100 - Londrina - PR**

CURSO DE APERFEIÇOAMENTO POR CORRESPONDÊNCIA

Solicito o mais rápido possível informações sem compromisso sobre o CURSO de . . . . . .

Nome . . . . . .

Rua . . . . . .

Cidade . . . . . .

Bairro . . . . . **CEP** . . . . . .

r0, r1, r2 — Número do registro a ser acessado no modo aleatório. Os bytes r0 e r1 são tratados como uma palavra de 16 bits, podendo assumir valores de 0 a 65535. O byte r2 é apenas usado para acumular o **vai um** (overflow).

Ao ser chamado para abrir um arquivo o BDOS passa a comparar os bytes de 0 a 14 do FCB com os bytes de mesmo número de cada entrada do diretório do drive indicado.

Ao encontrar uma entrada que "case" com o FCB, o BDOS copia os bytes de 15 a 31 desta entrada para as posições equivalentes no FCB. Porém, antes de iniciar a comparação, o BDOS salva o conteúdo do campo **dr** (byte 00) do FCB e põe em seu lugar o número do usuário corrente, a fim de possibilitar a comparação com o campo **us** do diretório. Após a busca, o BDOS restaura o valor deste campo.

Quando é chamado para fechar um arquivo, o BDOS efetua o teste do bit s2(7) do FCB, conforme mencionamos acima, e se necessário, salva o conteúdo das posições 15 a 31 do FCB nas mesmas posições da entrada do diretório.

AS TABELAS DE PARÂMETROS DE DISCO

Quando apresentamos a rotina SELDSK do BIOS, mencionamos que esta deveria retornar, no par HL, o endereço de uma tabela chamada **Disk Parameter Header**. Normalmente, cada drive de disco do sistema possui o seu cabeçalho de parâmetros de disco residente no BIOS. Estes cabeçalhos contém informações sobre o drive de disco a ele associado e áreas de trabalho utilizadas pelo BDOS durante o acesso a este drive.

DIRBUF — Endereço do **buffer** de diretório. Este **buffer** (128 bytes) é utilizado para armazenar os setores lidos ou escrever do/no diretório. Todos os cabeçalhos endereçam o mesmo **buffer de diretório.**

DPB — Endereço do bloco de parâmetros de disco (Disk Parameter Block — DPB). Este bloco (que será detalhado adiante) descreve as características de alocação de espaço do disco. Geralmente todos os DPH endereçam o mesmo DPB já que, na maioria dos casos, todos os drives de um sistema possuem as mesmas características lógicas.

CSV — Endereço da área de **check sum** do diretório. Na primeira vez que o BDOOS seleciona um drive, ele lê um determinado número de setores (definido no DBP — geralmente todos) do diretório, calcula o **check sum** de cada um e armazena o resultado nesta área (1 byte por setor). A partir deste instante, cada setor a ser inscrito na área de diretório pelo BDOS tem o **check sum** calculado e atualizado na área de **check sum** do drive ativo.

Agora, cada setor de diretório lido pelo BDOS tem o seu **check sum** calculado e comparado com o valor armazenado anteriormente. Em caso de diferença, o drive ativo é marcado como **READ ONLY** (somente de leitura). Isto evita que, inadvertidamente, o operador troque o disco de um drive durante a escrita em um arquivo, e o novo disco venha a sofrer gravações indevidas.

Quando uma escrita é tentada sobre um drive **read only**, o BDOS exibe a mensagem de erro:

Bdos Err On **d**: R/O

O estado de **R/O** é automaticamente retirado quando o sistema sofre uma partida a quente (Warm Start).

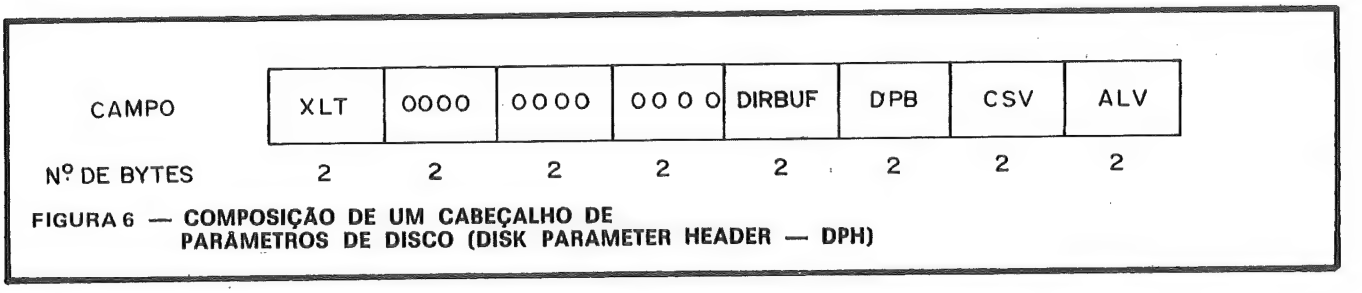


FIGURA 6 — COMPOSIÇÃO DE UM CABEÇALHO DE PARÂMETROS DE DISCO (DISK PARAMETER HEADER — DPH)

O DPH

A figura 6 mostra a composição de um cabeçalho de parâmetros de disco, cujos campo serão descritos agora:

XLT — Endereço da tabela de translação de setores mencionada quando apresentamos a rotina SECTRAN do BIOS. Os drives que utilizam o mesmo fator de entrelaçamento partilham a mesma tabela de translação. Este campo deve ser igual a 0000H se a translação de setores não é utilizada neste drive.

0000 — Estes 6 bytes são utilizados pelo BDOS para salvar valores intermediários utilizados durante o acesso ao diretório.

ALV — Endereço da área destinada à montagem da tabela de alocação de blocos. Esta tabela é utilizada pelo BDOS para indicar quais os blocos do disco que estão alocados. A cada bit desta tabela corresponde um bloco de disco. Se o bloco estiver alocado, o bit correspondente estará setado (= 1) e, caso esteja livre, o bit estará resetado (= 0). Ao selecionar um drive, o BDOS lê todas as entradas do diretório do disco nele existente, e seta na tabela de alocação os bits correspondentes aos blocos alocados para estas entradas. A partir daí, toda vez que o BDOS

precisa alocar um bloco ele procura seqüencialmente nesta tabela um bit = 0, e ao encontrar seta-o e aloca o bloco correspondente. De forma semelhante, quando um bloco deve ser liberado, o BDOS calcula a partir do número deste bloco a sua posição na tabela de alocação, e reseta o bit correspondente.

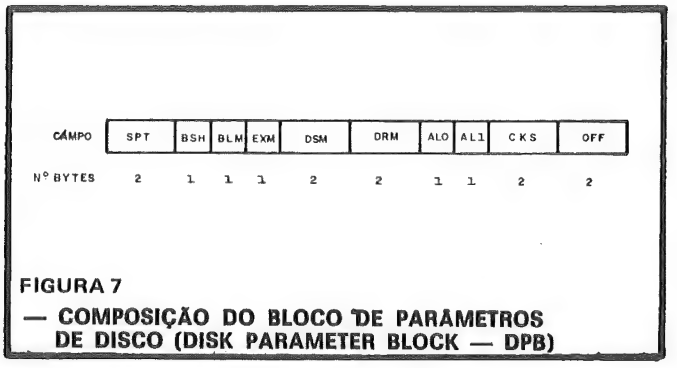


FIGURA 7
— COMPOSIÇÃO DO BLOCO DE PARÂMETROS DE DISCO (DISK PARAMETER BLOCK — DPB)

A figura 7 mostra a formação do bloco de parâmetros de disco, que será descrito agora:

SPT – Número de setores (de 128 bytes) por trilha.

BSH – Fator de deslocamento para alocação de blocos. Este parâmetro é utilizado pelo BDOS no cálculo do bloco relativo a um dado registro de um arquivo. O valor de BSH é determinado pelo seguinte método: BSH = log2 (No setores por bloco).

exemplo, suponhamos que utilizamos um bloco de 4 Kbytes e o nosso disco tenha menos de 256 blocos. Neste caso cada entrada do diretório terá capacidade de endereçar quatro extensões lógicas. Como poderíamos saber quais as extensões que se encontram em uma determinada entrada do diretório, se a mesma só possui um campo de ex? Este problema é resolvido fazendo-se uma operação lógica E entre o campo de ex e o complemento de EXM durante a comparação dos campos de ex do FCB e do diretório.

TAM. BLOCO (bytes)	DSM 256	DSM 256
1.024	0	—
2.048	1	0
4.096	3	1
8.192	7	3
16.384	15	7

No nosso exemplo, de acordo com a tabela acima. EXM será igual a 03H cujo complemento é FCH. Podemos então observar, que neste caso, somente os 6 bits mais significativos entram na comparação dos campos de ex, permitindo associar 4 extensões a cada entrada do diretório.

DSM – Número máximo de blocos que o disco pode conter, menos um. Este valor não inclui as trilhas de sistema.

FIGURA 8 — COMPOSIÇÃO DO CAMPO DE ALOCAÇÃO DE BLOCOS PARA DIRETÓRIO (NO DPB)

BLM – Número de setores por bloco menos um. Este parâmetro é utilizado pelo BDOS no cálculo da trilha e setores físicos correspondentes a um dado registro de um arquivo. A tabela abaixo mostra os valores para BSH e BLM em função dos tamanhos de blocos possíveis:

TAM. BLOCO (bytes)	BSH	BLM
1.024	3	7
2.048	4	15
4.096	5	31
8.192	6	63
16.384	7	127

EXM – Máscara para extensão. Esta máscara é utilizada pelo BDOS durante a comparação do campo ex do FCB com o campo ex de uma entrada do diretório. Isto é feito devido ao fato de que uma entrada de diretório pode, dependendo do tamanho de bloco utilizado, conter mais de uma extensão lógica. Por

DRM – Número máximo de entradas no diretório, menos um. Este valor determina o número máximo de arquivos que um disco pode conter.

ALO – Estes dois bytes são utilizados para reservar blocos para o diretório AL1.
Cada bit deste campo corresponde a um bloco do disco.

Durante a montagem da tabela de alocação de blocos, o BDOS move o conteúdo destes bytes para o início da referida tabela, onde o bit 00 corresponde ao bloco 00 e o bit 15 ao bloco 15.

O valor a ser colocado nestes bytes depende do número máximo de entradas no diretório e do tamanho de bloco utilizado. Suponhamos, por exemplo, que utilizemos blocos de 1 kbyte e 128 entradas no diretório. Neste caso, o diretório utilizará 4 blocos e os bits 00, 01, 02 e 03 de ALO deverão ser iguais a 1 e os demais iguais a 0.

CKS – Número de setores de diretório que têm o **check sum** calculado. Normalmente, no caso de discos flexíveis, verifica-se todos os setores do diretório. Nesta condição CKS

pode ser determinado pela fórmula abaixo:
CKS = (DRM = 1)/4

OFF – Número de trilhas iniciais reservadas para o sistema (COLD START + CCP + BDOS + BIOS). Normalmente são reservadas 2 trilhas.

O BDOS (Basic Disk Operating System) é o núcleo do sistema operacional CP/M. É ele o módulo responsável pelo gerenciamento, em alto nível, das operações de disco, console e impressora. As funções de alocação de espaço em disco; criação, apagamento,troca de nome, abertura e fechamento de arquivos; busca no diretório; escrita e leitura de registros em arquivos; etc., são todas executadas pelo BDOS. No tocante ao gerenciamento de console e impressora, o BDOS trata de funções do tipo: impressão de **string**, leitura do **buffer** de console, etc.

Como pode ser visto na figura 9, o BDOS ocupa cerca de 3,5 kbytes de memória. Da mesma, forma que o CCP (Console Command Processor) o BDOS pode, em caso de necessidade de muita memória, ser sobreposto pela área de dados de um programa transiente, desde que este não faça uso de nenhuma das funções do BDOS. Isto é possível dado ao fato de que a rotina WBOOT do BIOS recarrega o CCP e o BDOS. Muito raramente esta possibilidade é utilizada, já que a quase totalidade dos programas faz chamadas ao BDOS.

CHAMADA DAS FUNÇÕES

O BDOS (Versão 2.2) executa, ao todo, 39 funções distintas. Estas funções podem ser utilizadas pelos programas transientes através de um **CALL** para o vetor de entrada do BDOS, no endereço 0005H. Ao ser executado o CALL 0005H, o registrador C deve conter o número da função a ser executada e, o par DE, o parâmetro ou o endereço do parâmetro, de acordo com a função a ser executada. No caso da leitura seqüencial de um registro, por exemplo, o programa transiente deve passar 14H em C e, em DE, o endereço do FCB (File Control Block), o qual especifica o arquivo e o registro desejados.

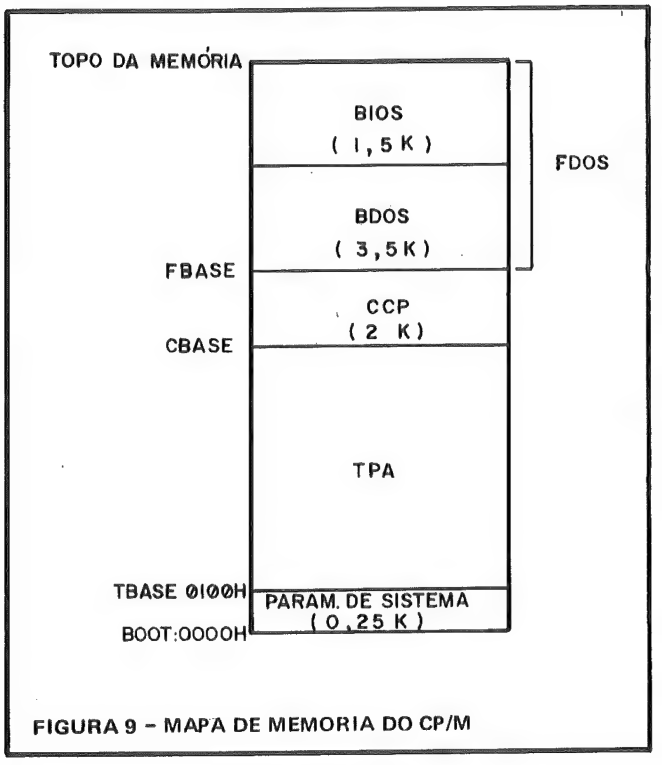


FIGURA 9 – MAPA DE MEMORIA DO CP/M

São as seguintes as funções executadas pelo BDOS (2.2):

Nº	FUNÇÃO
00	Reinicialização do sistema
01	Entrada de CONSOLE
02	Saída para CONSOLE
03	Entrada de **READER**
04	Saída para **PUNCH**
05	Saída para **LIST**
06	E/S direto para CONSOLE
07	Retorna IOBYTE
08	Atualiza IOBYTE
09	Imprime **string**
10	Lê o **buffer** de CONSOLE
11	Lê o status do CONSOLE
12	Retorna o número da versão
13	Reinicializa o sistema de discos
14	Seleciona disco
15	Abre Arquivo
16	Fecha Arquivo
17	Busca a primeira
18	Busca a próxima
19	Apaga arquivo
20	Lê seqüencialmente
21	Escreve seqüencialmente
22	Cria arquivo
23	Renomeia arquivo
24	Retorna o **Login Vector**
25	Retorna o disco corrente
26	Atualiza o **DMA ADDRESS**
27	Retorna o endereço do **ALLOCATION VECTOR**
28	Protege disco contra escrita
29	Retorna o **R/O VECTOR**
30	Altera os atributos do arquivo
31	Retorna o endereço do DPB
32	Lê/altera o código do usuário
33	Lê no modo aleatório
34	Escreve no modo aleatório
35	Calcula o tamanho do arquivo
36	Atualiza o RANDOM RECORD
37	Reinicializa um drive
38	Não implementada
39	Não implementada

Ao ser chamado, o BDOS salva o valor do **stack pointer**, substituindo-o pelo endereço do **stack** interno ao mesmo; salva o parâmetro passado em DE; verifica a validade do número da função passado em C e, caso este seja válido (0 a 40), passa a executar a função especificada. Do contrário, retorna imediatamente.

Ao retornar, o BDOS restaura o **stack pointer** e devolve o resultado ao código de errro da função executada no registrador A, caso este seja de um só byte ou no par HL, no caso de resultados de 2 bytes.

Por razões de compatibilidade com a linguagem de programação de sistemas PL/M da Intel, o BDOS retorna com o registrador A igual ao L e o registrador B igual ao H. Quando a função chamada não existe (> 40), os registradores A, B, H e L retornam com 00H.

AS FUNÇÕES

Apresentaremos agora cada uma das funções executadas pelo BDOS.

- **REINICIALIZAÇÃO DO SISTEMA (Função 0)**

A função reinicialização do sistema retorna o controle ao CCP (Console Command Processor). Quando esta função é chamada, o CCP e o BDOS são recarregados, a partir do disco A, pela rotina WBOOT do BIOS, e o controle é passado ao CCP, que seleciona física e logicamente o drive A, apresentando A > 7 no vídeo. Esta função faz exatamente o mesmo efeito que um JUMP para o endereço 0000H.

- **ENTRADA DE CONSOLE (Função 1)**

Esta função lê um caractere do console (teclado) e retorna com o mesmo no registrador A. Os caracteres imprimíveis (20H a 7EH), bem como RETURN (0DH), LINE FEED (0AH) e BACKSPACE (08H), são automaticamente ecoados para o console (vídeo). O caractere TAB (09H) é expandido em colunas de 8 caracteres, isto é, ao receber um TAB o BDOS envia para o console tantos espaços (20H) quantos forem necessários para fazer o cursor chegar à próxima coluna de tabulação. O BDOS interpreta, ainda, caracteres de parada do vídeo (ctl – s/13H) e ativação da cópia para impressora (ctl – p/10H). O controle só é devolvido ao programa chamador quando alguma tecla é pressionada.

- **SAÍDA PARA CONSOLE (Função 2)**

Envia para o console (vídeo) o caractere presente no registrador E. Da mesma forma que na função 1, os TABS expandidos. Antes de enviar o caractere em E para o console, o BDOS verifica, através da rotina CONST do BIOS, se existe algum caractere pendente no console (teclado). Caso exista, verifica se é um ctl – s ou ctl-p e, se for, trata-o da forma adequada. Quando o caractere pendente não é nem ctl-s nem ctl-p, o mesmo é armazenado para ser retornado na próxima chamada da função 01 – Entrada de console.

- **ENTRADA DE READER (Função 3)**

Lê um caractere de dispositivo físico associado ao dispositivo lógico **READER** e retorna no registrador A o código ASCII correspondente. O controle só é devolvido ao programa chamador quando um caractere é lido. Para executar esta função, o BDOS apenas chama a rotina READER no BIOS.

- **SAÍDA PARA PUNCH (Função 4)**

Envia o caractere presente no registrador E para o dispositivo físico associado à função PUNCH. Para executar esta função, o BDOS simplesmente chama PUNCH no BIOS.

- **SAÍDA PARA LIST (Função 5)**

Envia o caractere (ASCII) no registrador E para o dispositivo associado à função LIST (normalmente a impressora). De forma similar às funções 3 e 4, o BDOS chama diretamente a rotina LIST no BIOS.

- **E/S DIRETA PARA CONSOLE (Função 6)**

Conforme podemos observar na descrição das funções 01 e 02, o BDOS intercepta os caracteres ctl-s e ct-1p. Esta interceptação pode tornar-se inconveniente, por exem-

! CP/M no CP-500 !
! MUMPS !
! DBaseII !
! COBOL !
! Pascal UCSD !

Estes e mais dezenas de utilitários estão disponíveis para seu CP-500 com a instalação da placa CP/M, uma exclusividade da

$ Preços Promocionais $

CP/M ..... Cr$ 330.000
Expansão 64 K . Cr$ 275.000
Vídeo 80 col. . Cr$ 495.000
Drive Dupla Face (interno) Cr$ 1.250.000
Drive Dupla Face (externo) Cr$ 1.400.000

Ø Telematel Informática Ø

Rua Engº. Monlevade, 166 - São Paulo - SP - CEP 01308
Tel.:(011) 251-3430

Aceitamos representações para todo o Brasil - Preços válidos até 30/11/84 - Composição Ø Telematel Informática

plo, no caso de programas que geram gráficos animados. Para suportar aplicações desta natureza, o BDOS possui a função E/S DIRETA PARA CONSOLE.

Ao ser chamada esta função, o registrador E pode conter tanto o valor FFH, indicando uma requisição de entrada de console, como um caractere ASCII, indicando que o mesmo deve ser enviado para o CONSOLE (vídeo). Se for uma requisição de entrada de console, o caractere lido retorna no registrador A. Caso não exista caractere pendente no CONSOLE (teclado), esta função retorna de imediato com A = 00H.

- **RETORNA IOBYTE (Função 7)**

Esta função retorna, no registrador A, o valor do IOBYTE contido normalmente no endereço 0003H. (Vide a parte 2 deste artigo, caso necessite maiores detalhes sobre o IOBYTE).

- **ATUALIZA IOBYTE (Função 8)**

Esta função altera o IOBYTE para o valor contido no registrador E.

- **IMPRIME STRING (Função 9)**

Esta função envia para o console (vídeo) o **string** endereçado pelo par DE, até que um "S" seja encontrado no **string**. Da mesma forma que na função 02, ctl-s e ctl-p são detectados, e os TABS são expandidos.

- **LÊ O BUFFER DE CONSOLE (Função 10)**

A função de leitura de **buffer** de console põe no **buffer** endereçado pelo par DE um **string** teclado pelo usuário através do console. A entrada de console é terminada quando o usuário tecla RETURN ou quando o **buffer** enche. O **buffer** de console tem a seguinte forma.

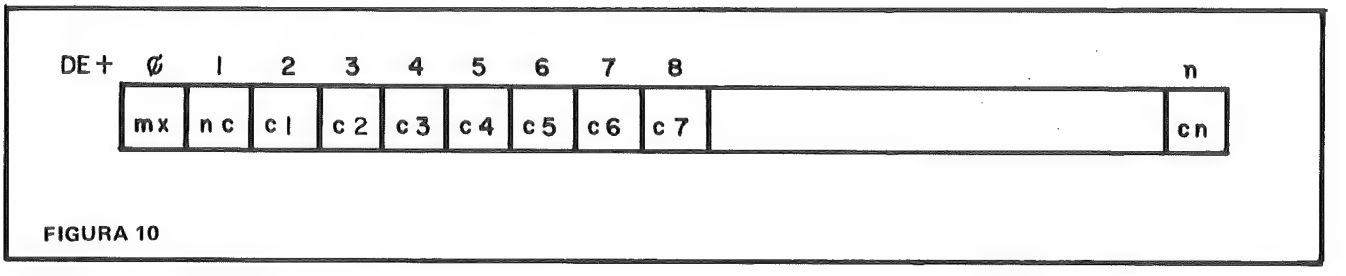


FIGURA 10

"mx" = número máximo de caracteres que o **buffer** pode conter (1 a 255), passado pelo programa chamador.
"nc" = número de caracteres lidos informado pelo BDOS ao retornar.
clacn = caracteres lidos.

Durante a edição de uma linha através desta função, são reconhecidos os seguintes caracteres de controle:

DELETE	(7FH)	remove do **buffer** e ecoa o último caractere teclado.
CTL–C	(03H)	reinicializa o sistema (WARM BOOT) quando teclado no início da linha.
CTL – E	(05H)	posiciona o cursor no início da próxima linha sem encerrar a edição
BACKSPACE	(08H)	volta o cursor uma posição, deletando o último caractere teclado.
LINE FEED	(0AH)	encerra a leitura da linha.
RETURN	(0DH)	encerra a leitura da linha.
CTL – R	(12H)	move o cursor para o início da próxima linha e reimprime tudo que já foi teclado.
CTL – U	(15H)	move o cursor para o início da próxima linha e desconsidera tudo que já foi teclado.
CTL – X	(18H)	retorna o cursor para o início da linha atual desconsiderando o que já foi teclado. O início mencionado refere-se à coluna na qual o cursor se encontrava quando a função 10 foi chamada.

- **PEGA O STATUS DO CONSOLE (Função 11)**

Esta função verifica se há algum caractere pendente no console (teclado). Se houver, o registrador A retorna com FFH. Do contrário, A retorna com 00H.

- **RETORNA O NÚMERO DA VERSÃO (Função 12)**

A função 12 retorna no registrador L o número da versão do CP/M. Para as versões anteriores a 2,0, o registrador L retorna com 00H. As versões, a partir da 2,0, retornam em L o número da versão em hexadecimal (20H para a versão 2,0, 22H para a 2,2 e assim por diante). Esta função tem por finalidade permitir que sejam escritos programas que rodem, sem problemas, tanto nas versões mais recentes como nas mais antigas. Por exemplo, pode-se escrever programas que acessem arquivos tanto no modo seqüencial como no aleatório e que, ao operar nas versões anteriores a 2,0 não façam uso do acesso aleatório, já que o mesmo não existe nestas versões.

CP/M 3.0

Ao começarmos a escrever este artigo, a última versão do CP/M era de 2,2 na qual nos baseamos até agora. A partir de janeiro de 1983, a Digital Research começou a distribuir, ainda de forma restrita, o CP/M 3.0 (CP/M PLUS).

Antes de prosseguirmos na descrição das funções executadas pelo BDOS, apresentaremos dois elementos importantes para o entendimento das explanações subseqüentes.

O LOGIN VECTOR

O BDOS mantém, internamente, um indicador de 16 bits chamado **LOGIN VECTOR**, destinado a indicar quais os drives que já foram selecionados desde que o equipamento foi ligado. No **LOGIN VECTOR** cada bit está associado a um drive (bit 0 – A, bit 1 – B, . . . , bit 15 – P). Na primeira vez que um drive é selecionado, o BDOS "seta" o bit

correspondente a este drive. Conforme poderemos constatar adiante, isto é feito, principalmente, para evitar que cada vez que um drive seja selecionado, o diretório do mesmo seja lido para efetuar a montagem do MAPA DE ALOCAÇÃO DE BLOCOS. Ao invés disto, ao selecionar um drive pela primeira vez, o BDOS monta na memória (no end. indicado no bloco de parâmetros de disco) o MAPA DE ALOCAÇÃO DE BLOCOS (ALV) correspondente ao drive selecionado e passa a utilizar-se do mesmo, sempre que necessita alocar ou liberar blocos neste drive.

O READ ONLY VECTOR

De forma semelhante ao **login vector**, o BDOS mantém um outro indicador de 16 bits, chamado **READ ONLY VECTOR**, destinado a indicar quais os drives que estão marcados como "só de leitura" (read only). Da mesma forma que no **login vector** cada bit corresponde a um drive (bit 0 – A, bit 1 – B, . . . , bit 15 – P) e, estando "setado" indica que o drive associado não pode receber uma operação de escrita.

Já foi mencionado anteriormente, que todos os setores do diretório (ou uma parte destes) têm o seu **checksum** calculado a fim de detectar uma troca de disco (mídia). Pois bem, quando esta troca de disco é detectada, o bit do **READ ONLY VECTOR** corresponde ao drive no qual ocorreu a troca é "setado". Por outro lado, antes de efetuar uma escrita em qualquer drive, o BDOS verifica o estado do bit do **READ ONLY VECTOR** correspondente ao drive desejado. Se este estiver "resetado" a escrita é efetuada normalmente. Do contrário, o BDOS não escreve e exibe a mensagem:

Bdos Err On d: R/O

Tanto o **READ ONLY VECTOR** como o **LOGIN VECTOR** são zerados quando o sistema é inicializado ou sofre uma partida a quente (WARMBOOT) e podem ser acessados por programa através de funções que serão mostradas adiante.

Esta função é normalmente utilizada para permitir a troca de discos (mídia) durante a execução de um programa, sem que o drive correspondente seja marcado como **READ/ONLY**.

- **SELECIONA DISCO (Função 14)**

A presente função designa o drive de disco cujo número se encontra no registrador E (E = 0 para A, E = 1 para B, . . . E = 15 para P) para ser utilizado como **default** nas próximas operações de disco. Ao executar esta função, o BDOS verifica inicialmente se o drive especificado já está logicamente selecionado e caso esteja, retorna ao programa chamador. Do contrário, chama SELDSK no BIOS e, de posse do endereço do BLOCO DE PARÂMETROS DE DISCO (DPH/DPB) do drive selecionado, move o conteúdo deste para uma área interna (ao BDOS) a fim de facilitar o acesso a estes dados. Feito isto, o BDOS verifica o estado do bit correspondente ao drive selecionado no **LOGIN VECTOR** e, se o mesmo estiver "setado", retorna de imediato ao programa chamador. Do contrário, "seta" este bit e lê todas as entradas do diretório a fim de montar a TABELA DE ALOCAÇÃO DE BLOCOS (ALV), "setando" os bits da tabela correspondentes aos blocos alocados pelas entradas.

Nas próximas operações com arquivos em que o campo "dr" do FCB for igual a 0 o BDOS assumirá o drive indicado através desta função. No entanto, para "dr" entre 1 e 16, será utilizado o drive especificado (A = 1, B = 2, . . . P = 16).

- **ABRE ARQUIVO (Função 15)**

A função "abre arquivo" tem por objetivo ativar um arquivo que já existe no diretório, cujo FCB é endereçado pelo par DE. Para executar esta função, o BDOS substitui, temporariamente, o conteúdo do campo "dr" do FCB pelo número do usuário ativo, a fim de que só os arquivos pertencentes a este sejam considerados. Feito isto, o BDOS passa a comparar os bytes de 0 a 14 (μs a s2) do FCB com

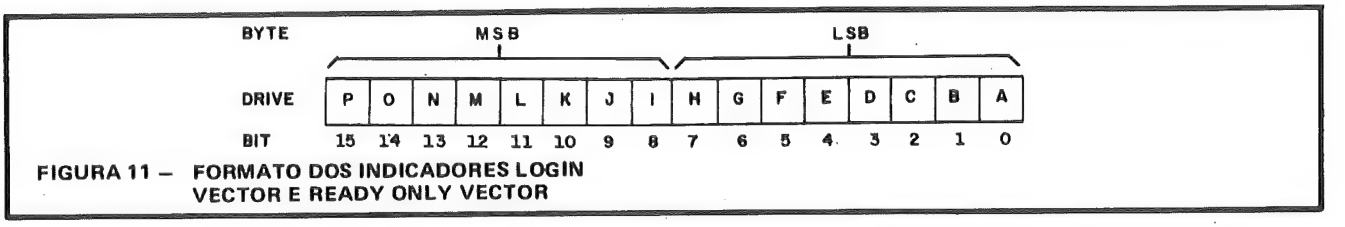


FIGURA 11 – FORMATO DOS INDICADORES LOGIN VECTOR E READY ONLY VECTOR

Prosseguiremos agora a descrição das funções executadas pelo BDOS.

- **REINICIALIZA O SISTEMA DE DISCOS (Função 13)**

Esta função é utilizada para, através de programa, levar o sistema de arquivos ao estado inicial de quando a máquina foi ligada. Para efetuar isto, o BDOS zera todos os bits do **login vector** e do **read only vector**, permitindo assim que todos os discos possam ser escritos (read/write), seleciona o drive A e assume o endereço de **default** para o buffer de disco (DMA ADDRESS = 0080H).

Não é aconselhável chamar esta função com algum arquivo em aberto pois, se o mesmo sofreu alguma escrita durante o período em que esteve aberto, os dados gravados serão perdidos, já que o BDOS só atualiza o diretório quando o arquivo é fechado.

os bytes correspondentes de cada entrada do diretório, até encontrar uma que "case" Durante esta pesquisa, a presença de um "?" (ASCII 3FH) em qualquer um dos bytes de 1 a 14 do FCB (vide referência ambígua na parte 1) faz com que este byte seja irrelevante na comparação.

Ao encontrar uma entrada de diretório que "case" com o FCB, o BDOS copia o conteúdo das posições "d0" a "d15" (Nºs dos blocos alocados) da entrada para as mesmas posições no FCB e "seta" o bit 7 do campo "s2" que é "resetado" quando o arquivo é escrito. Feito isto, o controle é devolvido ao programa chamador com o registrador A contendo um valor de 0 a 3 (posição relativa da entrada no buffer de diretório), se a abertura teve sucesso, ou 0FFH, se o arquivo não pode ser encontrado.

É importante observar que o ponto onde o arquivo será aberto é determinado pelo conteúdo dos campos "ex"

e "s2", que determinam a extensão a ser acessada, e do campo "cr", que indica a posição do registro na extensão. Normalmente, quando o arquivo vai ser acessado seqüencialmente, estes campos são zerados a fim de que o arquivo seja aberto no primeiro registro. Quando o arquivo vai ser acessado aleatoriamente, é necessário que estes campos sejam zerados. O campo s1 é automaticamente zerado pelo BDOS.

- **FECHA ARQUIVO (Função 16)**

A função "fecha arquivo" executa o inverso da função "abre arquivo". Ela tem por objetivo gravar no diretório do disco as informações relativas à alocação de espaço contidas no FCB endereçado por DE, o qual, admite-se, corresponde a um arquivo previamente aberto. Da mesma forma que para a função "abre arquivo", o BDOS "varre" o diretório do disco especializado no FCB em busca de uma entrada que "case" com o FCB. Ao encontrar, o BDOS move o conteúdo dos campos "d0" a "d15" do FCB para as posições correspondentes da entrada de diretório.

Antes de executar as operações acima, o BDOS verifica o bit 7 do campo "s2" do FCB, a fim de determinar se o arquivo foi escrito durante o tempo em que esteve aberto. Caso este não tenha sido escrito (bits2 (7) = 1), o controle é devolvido, de imediato, ao programa chamador. Do contrário, as operações já mencionadas são executadas. Isto é feito com o intuito de minimizar o acesso ao diretório.

Ao retornar, o registrador A contém um valor entre 0 e 3 (posição relativa da entrada no buffer de diretório) caso o fechamento tenha sido efetuado com sucesso ou, 0FFH, se o arquivo não foi encontrado no diretório.

- **BUSCA A PRIMEIRA (Função 17)**

Esta função "varre" o diretório, desde o início, em busca de uma entrada que "case" com o FCB endereçado pelo par DE. Caso o arquivo desejado seja encontrado, o **buffer** de disco (DMA ADDRESS) é preenchido com o setor do diretório que contém a entrada que coincidiu, e o registrador A retorna com um valor de 0 a 3, indicando a posição relativa da entrada no **buffer** de disco. Do contrário, ou seja, se o arquivo não for encontrado, o registrador A retorna com 0FFH.

Já que cada entrada ocupa 32 bytes, para calcular o endereço absoluto da entrada no **buffer** de disco basta multiplicar o conteúdo de A por 32 (ADD A 5 vezes) e adicionar o resultado ao endereço inicial do **buffer** (DMA ADDRESS).

Da mesma forma que na função **"abre arquivo"**, a presença de um "?" (ASCII 3 FH) em qualquer um dos bytes de 1 a 14 do FCB faz com que este byte seja irrelevante na comparação. A presença de um "?" no campo "dr" (byte 0) do FCB, faz com que o drive de **default** seja utilizado e o campo "us" (nº do usuário que criou o arquivo) das entradas do diretório seja eliminado da comparação.

Esta função, normalmente, não é utilizada por programas aplicativos. A título de exemplo, podemos citar o utilitário STAT, que se utiliza desta função para calcular, a partir das informações contidas no diretório, o tamanho de um arquivo.

- **BUSCA A PRÓXIMA (Função 18)**

A função "busca a próxima" é similar à anterior, exceto que a pesquisa começa a partir da última entrada acessada pelo BDOS. Geralmente, esta função é utilizada quando, após ter-se encontrado uma entrada de um dado arquivo, utilizando-se a função 17 (BUSCA A PRIMEIRA), deseja-se verificar se há uma outra entrada deste mesmo arquivo.

- **APAGAR ARQUIVO (Função 19)**

Esta função retira do diretório as entradas correspondentes ao(s) arquivo(s) especificado(s) no FCB endereçado por DE. Para executar esta função, o BDOS "varre" o diretório em busca de entradas que "casem" com o FCB e, ao encontrá-las, põe no campo "us" destas entradas o valor E5H. Além disto, o BDOS libera, no MAPA DE ALOCAÇÃO DE BLOCOS (ALV) do drive especificado, os blocos anteriormente alocados por esta entrada.

A presença, agora, do valor E5H no campo "us" da entrada passa a indicar que esta está livre, podendo ser utilizada para outros arquivos. A escolha do valor E5H deve-se ao fato de que, no padrão IBM de formatação de discos flexíveis, os setores são inicializados com este valor. Isto simplifica bastante a formatação de discos, evitando a necessidade de uma formação inicial do diretório.

Como podemos observar, tanto a(s) entrada(s) no diretório como os dados contidos no arquivo que foi "deletado" permanecem intactos no disco. Esta peculiaridade é utilizada por programas que recuperam arquivos apagados acidentalmente como, por exemplo, o utilitário RECDIR. COM do DGT-101 da DIGITUS. De forma simplificada, estes programas agem da seguinte forma:

> *Acham no diretório a(s) entrada(s) correspondente(s) ao arquivo desejado.*
> *De posse dos números dos blocos alocados pelo arquivo (campo "d0" a "d15" da(s) entrada(s)) verificam, no MAPA DE ALOCAÇÃO DE BLOCOS, se estes já foram alocados por outros arquivos.*
> *Se algum dos blocos já foi realocado, o arquivo se encontra, pelo menos parcialmente, destruído e o operador é informado. Do contrário, estes blocos são novamente alocados e a(s) entrada(s) reativada(s) substituindo-se o valor E5H do campo "us" pelo número do usuário corrente.*

Os campos de nome e tipo (bytes 1 a 11) do FCB podem conter uma referência ambígua (presença de "?"), possibilitando o apagamento de diversos arquivos de uma só vez. No entanto, o campo "dr" deve, obrigatoriamente, especificar o drive a ser utilizado (0 = **default**).

Em caso de sucesso ao retornar, o registrador A contém um valor entre 0 e 3 (posição relativa da entrada no setor de diretório). Caso o(s) arquivo(s) não tenha(m) sido encontrado(s), A retorna com FFH.

- **LÊ SEQUENCIALMENTE (Função 20)**

Assumindo-se que o FCB endereçado por DE refere-se a um arquivo previamente aberto (pelas funções 15 a 22), esta função lê, do arquivo para o **buffer** de disco, (DMA ADDRESS) o registro (128 bytes) especificado pelo campo "cr". Se após o incremento, "cr" "estoura" (= 128), a próxima extensão deste arquivo (se houver) é automaticamente aberta e o campo "cr" é zerado, habilitando o acesso ao próximo registro do arquivo.

Note que, neste caso, na próxima operação de leitura, se o programa chamador não alterou o conteúdo de "cr"

(há restrições para esta alteração), o registro imediatamente seguinte será lido, fazendo-se assim um acesso estritamente seqüencial.

Em caso de sucesso, ao retornar o registrador A contém 00H, enquanto FFH é retornado quando chega-se ao fim (físico) do arquivo.

- **ESCREVE SEQUENCIALMENTE (Função 21)**

Esta função é em tudo análoga à função 20, a que lê seqüencialmente. Tal como ela, esta função requer alguns preparativos: primeiro, é necessário que tenhamos montado um FCB (File Control Block) e o tenhamos transmitido à função do BDOS para abrir arquivos (a de número 15) ou à função para criar arquivos (a de número 22). Além disso, é preciso também ter estabelecido a área de memória a ser usada pelo DMA, o que é feito usando-se a função 26. Considerando isso, o uso desta função é simples: passamos no par de registradores DE o endereço do FCB do arquivo e chamamos o BDOS (CALL 5) com o valor 21 decimal (ou 15H) no registrador C.

Após cada chamada desta função, o contador de registros seqüencial é incrementado de um até atingir um máximo de 128 registros (setores). No caso de se continuar a escrever no arquivo o BDOS abrirá uma nova extensão, registrando este fato no FCB do arquivo.

O registrador A deverá retornar normalmente o valor zero. Qualquer outro valor significa que a operação de escrita não pode ser efetuada devido à falta de espaço no disco. Este é o único erro que pode ser detectado por programa; todos os outros erros de hardware são tratados diretamente pelo BDOS ou pelas rotinas do BIOS.

- **CRIAR ARQUIVO (Função 22)**

Esta função cria um arquivo com o nome e o tipo especificado. A chamada é feita da maneira usual: endereço do FCB no par de registradores DE e o código da função (22D ou 16H) no registrador C. O BDOS retorna no registrador A o código do diretório referente ao arquivo. Como vimos, este código é o número relativo do arquivo no diretório, isto é, 0 para o primeiro arquivo, 1 para o segundo, 2 para o terceiro e assim por diante. O único tipo de erro que pode ser detectado por esta função é o caso de disco ou diretório cheio, o que acarretará que o registrador A volte com o valor 0FFH (255 decimal).

É preciso tomar algumas precauções antes de se chamar esta função. Em primeiro lugar é preciso se certificar que não existe nenhum arquivo com o mesmo nome no diretório. Isso pode ser conseguido de duas maneiras: ou tentando abrir o arquivo e verificando se a tentativa teve sucesso, ou então apagando incondicionalmente o arquivo (chamada número 19). Outro detalhe a ser observado é que esta função não faz qualquer tipo de consistência sobre o nome do arquivo colocado no FCB. Em conseqüência disso, se pusermos um nome escrito com letras minúsculas por exemplo, não poderemos mais acessar este arquivo por comandos do CCP. Por outro lado, isto pode ser até um efeito desejável, quando queremos criar um arquivo que não possa ser apagado pelo comando ERA, por exemplo.

- **REBATIZAR ARQUIVO (Função 23)**

Esta função serve para trocar o nome de um arquivo no diretório. O modo de chamar esta função é um pouco estranho: o PCB cujo endereço é passado nos registradores D

CARACTERÍSTICAS ESPECIAIS

- **Teclado** Auto-repetitivo Maiúsculas e minúsculas diretamente. Sem contatos, indutivo com Control/Reset
- **Fonte de Alimentação** Chaveada 5 amperes com proteção contra curto circuito e sobre tensões 110/220 diretos
- **Garantia** 6 meses total
- **Manual** Ilustrado com 380 páginas editado pela editora MC. Graw-Hill do Brasil.
- **Saída Som direto pela TV.**
- **Sacola vinílica para transporte**

DM Eletrônica Ltda

Rua Campevas, 86 - casa 1
Tel.: 864-7561 - Perdizes - SP.

Representantes

Tiger - Av. Rebouças, 3199 - SP
Fotoptica - Av. Rebouças, 2315 - SP
Microshop - Alameda Lorena, 652 - SP

e E contém o nome do arquivo a ser rebatizado nos 16 primeiros bytes e o novo nome dos 16 bytes seguintes. O registrador A volta com um código de diretório normal no caso do nome antigo ter sido encontrado ou 0FFH, em caso contrário.

Observe que o BDOS verificará apenas se há efetivamente o arquivo a ser rebatizado, porém não procurará no diretório um arquivo com o novo nome. O usuário deverá tomar a precaução de se certificar disso, tentanto abrir um arquivo com o novo nome e verificando se ele já não existe.

Assim como na função de criar arquivo, os nomes do arquivo são tomados literalmente e, portanto, denominações ambíguas de arquivos não são permitidas. Qualquer tentativa nesse sentido produzirá resultados imprevisíveis.

- **RETORNA O LOGIN VECTOR (Função 24)**

Esta função retorna no par de registradores HL o mapa de discos ativos (LOGIN VECTOR), que indica quais drives foram acessados desde a última reinicialização do sistema (WARM BOOT) ou desde a última chamada da função 13 (a que reinicializa os discos). O Bit 0 do registrador L refere-se ao drive A, enquanto que o bit 7 do registrador H refere-se ao drive P. Se o bit vale 1, isto significa que o drive correspondente já foi selecionado.

- **RETORNA O DISCO CORRENTE (Função 25)**

Esta função retorna em A o código do disco **default** sendo usado, isto é, aquele que foi indicado na última chamada à função número 14 ou então o que foi indicado por um comando X: do CCP, onde X vale A, B, C. . . P. O valor do acumulador é 0 se o disco em questão é o A, 1 se é o B, e assim sucessivamente.

A utilidade desta função reside principalmente quando desejamos formatar o nome de um arquivo a partir do FCB, a fim de mostrá-lo na tela ou em qualquer outro dispositivo. Isto pode ser feito facilmente somando-se o valor 65 ou 41H ao valor retornado pela função, dando, portanto, o caractere ASCII correspondente ao nome do disco.

- **ESTABELECE O ENDEREÇO DA ÁREA DE DMA (Função 26)**

Esta função é usada para estabelecer o endereço a ser usado pelo sistema para a operação de "direct memory access" quando da leitura ou escrita de dados no disco. O endereço é passado como de praxe no par de registradores DE.

Na verdade, nem sempre esta função faz realmente o que o nome indica. Mais freqüentemente, ela serve apenas para indicar o endereço do buffer de leitura/escrita – o endereço real do DMA é estabelecido pelo hardware, geralmente em alguma área obscura da memória.

- **RETORNA VETOR DE ALOCAÇÃO (Função 27)**

Esta função retorna o endereço do vetor de alocação do disco sendo usado no momento. Este valor consiste basicamente de uma tabela de bits, cada um correspondente a uma unidade de alocação, isto é, 1 Kbyte de espaço em disco (em versões anteriores do CP/M uma unidade de alocação correspondia a 2 Kbytes). Esta rotina praticamente não é utilizada por programas aplicativos feitos pelo usuário, no entanto, alguns utilitários do próprio CP/M o usam, como por exemplo o programa STAT.

A Digital Research considera o formato do vetor de alocação como uma informação confidencial, porém, algumas informações a respeito podem ser encontradas num manual fornecido pela própria Digital Research, chamado "CPM Alteration Guide".

Para quem tem intenção de se valer desta função do BDOS, um cuidado deve ser tomado: o endereço do vetor de alocação pode vir errado caso o disco sendo acessado no momento esteja protegido contra escrita (estado Read Only).

- **PROTEGE DISCO CONTRA ESCRITA (Função 28)**

O propósito desta função é proteger o disco presentemente sendo utilizado contra escrita, isto é, ela liga o bit correspondente ao disco no READ ONLY VECTOR. Qualquer tentativa posterior de se escrever no referido disco será interceptada pelo BDOS que, então, emitirá no console uma mensagem como a seguinte:

BDOS Err on X: R/O

onde X: é o nome do disco selecionado.

Convém observar que o estado de READ ONLY permanece, mesmo quando se seleciona um outro disco. De fato, a única maneira de se desproteger o disco contra escrita, subseqüentemente, é através de uma partida a frio ou a quente (Cold ou Warm Start).

- **RETORNA O READ ONLY VECTOR (Função 29)**

Esta função retorna o READ ONLY VECTOR descrito anteriormente no par de registradores HL. Neste mapa de bits, os bits 0, 1, 2, etc. . . correspondem aos discos A, B, C . . . P. Como também já foi explicado, o CP/M liga o bit do LOGIN VECTOR correspondente ao disco em questão quando este é protegido por uma chamada à função 28 do BDOS ou, quando o próprio BDOS descobre que o disco foi trocado.

- **ESTABELECE ATRIBUTOS DE ARQUIVOS (Função 30)**

Esta função serve para alterar o estado dos chamados atributos de um determinado arquivo, dentro do diretório do disco. Esses atributos são armazenados dentro do diretório, utilizando-se os bits mais altos de cada um dos 11 caracteres que compõem o nome do arquivo. Isto é possível, uma vez que o código ASCII utilizado pelo CP/M só precisa dos 7 bits mais baixos dos caracteres, deixando assim um bit livre que pode ser usado para armazenar um "flag" qualquer. Chamaremos aos 8 bits correspondentes aos 8 caracteres do nome do arquivo, respectivamente, de F1, F2, F3 . . .F8; e aos 3 bits correspondentes aos 3 caracteres da extensão de T1, T2 e T3. Convencionado isto, podemos relacionar esses bits com os atributos de arquivo da seguinte forma:

F1 a F4 – > Livres para serem usados
F5 a F8 – > Reservados para uso futuro do CP/M
T1 – > Atributo "READ ONLY"
T2 – > Atributo "Arquivo do Sistema"
T3 – > Reservado para uso futuro do CP/M

Vemos portanto que dos 11 "flags" disponíveis, o CP/M utiliza 2, reserva 5 para uso futuro e deixa 4 livres para serem empregados pelo usuário. O atributo "READ ONLY" serve, como seria de se esperar, para marcar um determinado arquivo como protegido, isto é, ele não pode ser escrito ou apagado. O atributo "Arquivo do Sistema" serve presentemente apenas para que o comando DIR do CCP omita esses arquivos quando da listagem do diretório do disco. Os "flags" livres podem ser utilizados pelo usuário

para qualquer fim, como por exemplo, para marcar certos arquivos como sendo de segurança.

A maneira de se invocar esta função se enquadra nos mesmos padrões utilizados até agora, ou seja, passa-se no par de registradores DE o endereço de um FCB que já deverá ter acoplado nos caracteres correspondentes ao nome do arquivo os atributos que se deseja modificar. O BDOS encarrega-se então de procurar no diretório o arquivo corrrespondente, atualizando-o. Também como de hábito, o registrador A retorna o código do diretório.

- **RETORNA O DISK PARAMETER BLOCK (DPB) (Função 31)**

Esta função não é, normalmente, chamada por programas aplicativos feitos por usuários. Quando invocada, retorna no par de registradores HL o endereço do bloco de parâmetros de disco (DPB). Esta área de 15 bytes contém informações relativas ao formato do disco correntemente selecionado, tais como: quantidade de setores por trilha, capacidade do disco, número de trilhas existentes antes da trilha do diretório, etc. . .

- **LÊ OU MODIFICA O NÚMERO DO USUÁRIO (Função 32)**

O número do usuário é um valor entre 0 e 15, que afeta a localização de um arquivo durante uma pesquisa de diretório. Esta função é usada geralmente quando mais de um usuário costuma usar um mesmo arquivo. Para se consultar o número do usuário corrente coloca-se o valor FF hexadecimal no registrador E. Após o retorno da função, o registrador A conterá o número do usuário corrente. Se, ao contrário, quisermos modificar o número do usuário colocamos o valor desejado em E antes de chamar a função.

- **LÊ RANDOMICAMENTE (Função 33)**

Esta função lê um registro específico (128 bytes) de um arquivo randômico, isto é, um arquivo em que os registros podem ser acessados em qualquer ordem. Para usar esta função assume-se que o usuário já tenha aberto o arquivo, especificado o endereço de DMA (através da função número 15 do BDOS). Ao chamar a função, o endereço do FCB do arquivo deverá ser passado como de hábito no par de registradores DE. O número do registro a ser lido é um valor de 24 bits discriminado nas posições relativas do FCB números 33, 34 e 35. O byte 33 contém a parte baixa o byte 34 a parte média e o byte 35 a parte alta. O CP/M utiliza a parte alta (byte 35) unicamente para calcular o tamanho global do arquivo quando é chamada a função 35. Este byte deve ser zerado quando se deseja apenas uma leitura randômica. Isto nos deixa com um valor de apenas 16 bits para o número do registro, isto é, podemos acessar desde o registro 0 até o registro 6535, perfazendo um total de 8.388.480 bytes para o tamanho máximo de um arquivo.

Ao retomar da função, o BDOS coloca em A um código de erro que pode ser interpretado da seguinte maneira:

A = 0 – > Operação terminada com êxito.
A = 1 – > Tentativa de leitura de um registro não escrito
A = 3 – > O CP/M não pode fechar a extensão corrente
A = 4 – > Tentativa de leitura com extensão não aberta
A = 6 – > Tentativa de leitura além do final do disco

Os erros 1 e 4 ocorrem quando tentamos ler um registro que não foi escrito. O erro 3 não ocorre quando o sistema é utilizado corretamente, e pode ser sanado enviando-se um comando de leitura na extensão 0 do arquivo – este tipo de erro ocorre quando um processo de leitura randômica é iniciado sem se ter acessado previamente a extensão 0 do arquivo (esta é uma precaução fundamental quando se trata arquivos randômicos). O erro 6 acontece quando o byte 35 do FCB é diferente de 0.

Note que, ao contrário do que acontece quando lemos um arquivo seqüencialmente, após uma leitura randômica o contador seqüencial de registros do FCB não é atualizado, isto é, qualquer operação de leitura ou gravação subseqüente se fará a partir deste mesmo registro. De fato, é possível seguir uma operação de leitura seqüencial com um comando de gravação seqüencial, por exemplo – isto faria com que o registro lido fosse regravado e o número de registro seqüencial fosse incrementado, de maneira que uma próxima gravação seqüencial se faria no registro seguinte. Já, se escolhêssemos seguir uma leitura randômica com uma leitura seqüencial, esta última iria ler exatamente o mesmo registro lido randomicamente.

- **GRAVA RANDOMICAMENTE (Função 34)**

Esta função grava um registro (128 bytes) numa posição qualquer de um arquivo randômico. Os preparativos para o uso desta função são exatamente os mesmos da função 32 (lê randomicamente); devemos primeiro abrir o arquivo e estabelecer o endereço de DMA para apontar os 128 bytes que queremos escrever no arquivo, além de indicar nas posições 33 e 34 do FCB o número do registro a ser escrito. Quando a extensão correspondente ao registro não foi ainda acessada, uma nova extensão é criada automaticamente.

Os códigos de erro devolvidos no registrador A são exatamente os mesmos da função de leitura randômica, acrescentando-se o código 5, indicando diretório cheio.

Esta função também não incrementa o contador de registros seqüencial, de maneira que qualquer função de leitura ou gravação seqüencial acessará o mesmo registro que foi gravado randomicamente.

COMPUTRON

MICROS
• TODAS AS MARCAS
• NOVOS • CONSIGNAÇÃO
• USADOS • SUPRIMENTOS

CURSOS
• BASIC
• VISICALC • ASSEMBLER
• DBASE II • APLICATIVOS

CONSULTORIA
• SOFTWARE
• HARDWARE • IMPLANTAÇÕES
• SISTEMAS • PROJETOS

NUM SÓ LUGAR TUDO O QUE VOCÊ PRECISA!
Rua Pamplona, 818 - 1º andar - Jardim Paulista - São Paulo • R. Vergueiro, 2551 - Vila Mariana - São Paulo

A precaução de abrir a extensão 0 do arquivo antes de efetuar uma seqüência de gravações randômicas também deve ser observada.

- **RETORNA O TAMANHO DO ARQUIVO (Função 35)**

Esta função recebe no par de registradores DE o endereço de um FCB e retorna neste mesmo PCB, nas posições relativas 33, 34 e 35 o número do último registro do arquivo somado de 1. Os bytes 33 e 34 são, respectivamente, a parte mais baixa e mais alta desse número, sendo que o byte 35 só voltará com um valor diferente de 0 se o arquivo possuir exatamente 65,536 registros, o máximo permitido.

Se o BDOS não encontrar o arquivo correspondente ao FCB, este retorna com as posições 33 e 34 valendo 0.

Esta função é útil quando desejamos acrescentar dados ao final de um arquivo. Se após sua chamada invocarmos a função de gravação randômica, o novo registro será posto no final do arquivo.

É preciso que se saliente que esta função não devolve na verdade o tamanho do arquivo, e sim o número do registro posterior ao último. Assim, um arquivo randômico que possua gravados apenas o registro 0 e 5000, por exemplo, devolverá o valor 5000 como resultado, quando em verdade o tamanho real do arquivo é muito menor.

- **ESTABELECE O NÚMERO DE REGISTRO RANDÔMICO (Função 36)**

Esta função coloca na parte do FCB reservada para o número de registro randômico o valor correspondente ao último registro lido ou escrito seqüencialmente.

Esta função pode ser utilizada, por exemplo, para criar um arquivo indexado. Basta que, a cada registro que se escreve seqüencialmente num arquivo, chamemos esta função e guardemos o número de registro randômico num outro arquivo. Este último arquivo poderá então servir de arquivo índice para se acessar registros do primeiro arquivo.

- **REINICIALIZA O DISCO LÓGICO (Função 37)**

Esta é uma versão mais específica da função 13, a que reinicializa o sistema de discos, na medida em que permite que se especifique um disco em particular ao invés de todos eles.

Em DE passamos um mapa de bits onde especificamos quais discos deverão ser reinicializados. O bit mais alto de D corresponde ao drive P:; enquanto que o bit mais baixo de E refere-se ao drive A:. A função reinicializará apenas os drives cujos bits respectivos possuam o valor 1.

Quando do retorno da função, o valor do registrador A será sempre 0.

- **GRAVA RANDOMICAMENTE REGISTROS COM ZEROS (Função 40)**

Esta função foi incluída no CP/M para auxiliar a firma MICROSOFT a implementar seu compilador COBOL. Seu objetivo é preencher rapidamente um arquivo com registros vazios, geralmente para possibilitar um processo posterior de leitura/gravação randômica. Só é preciso invocar esta função uma vez para preencher todo bloco de alocação correspondente ao registro especificado com zeros, isto é, considerando-se que um bloco de alocação possui 1k (8 registros), para preencher um arquivo com um tamanho de 10k precisamos chamar esta função apenas 10 vezes.

Para usar esta função adota-se os mesmos procedimentos empregados na função 34.

DIGA ADEUS
AS INTERFACES

ADQUIRA JÁ O MAIS NOVO LANÇAMENTO DO MERCADO NACIONAL.

APPLE MASTER

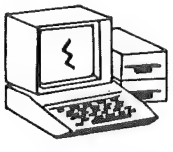
* CPU COM 64KBYTES
CPM RESIDENTE
TECLADO PROGRAMAVEL COM
MAIUSCULAS E MINUSCULAS.

E AINDA COM A GARANTIA DE QUEM
FABRICA O APPLE II PLUS

■ PROMOÇÃO PARA NOVEMBRO ■

01 APPLE MASTER
01 MONITOR DE VIDEO
02 DISK DRIVE
01 INTERFACE P/ DRIVE

APENAS CR$ 5.630.000,-



SIGMATRON
COMÉRCIO DE COMPUTADORES LTDA.

AV. COTOVIA, 350 - CEP 04062 - FONE: 240-9004 - INDIANÓPOLIS - SP

UTILIZANDO AS FUNÇÕES BDOS

Iremos agora examinar alguns aspectos práticos da utilização das funções do BDOS. Para tanto, o melhor mesmo é elaborar um programa que as utilize. O uso de linguagem Assembler será, é claro, uma necessidade. O leitor que não estiver familiarizado com Assembler 8080 não deve continuar adiante, sem antes dar ao menos uma olhada num manual ou livro que trate do assunto.

O programa que escolhi como exemplo não é nada de extraordinário, no entanto, contém procedimentos que ilustram muito bem a maioria dos procedimentos usuais no trato com o BDOS. Além disso, este programa pode ser usado como subrotina em outros programas utilitários. Trata-se de um programa que executa a mesma função do comando intrínseco "DIR", isto é, serve para mostrar na tela os arquivos contidos num determinado disco.

Para montar e obter a listagem que apresentamos a seguir foi utilizado o Assembler mais comum em sistemas CP/M – o "ASM", da Digital Reserarch.

ENTENDENDO O PROGRAMA

Para atuar de maneira semelhante ao comando DIR, nosso programa precisa considerar qual disco e arquivos se deseja pesquisar, ou seja, precisamos utilizar a especificação de arquivo citada em seguida ao nome do programa. Quando dizemos.

DIR *.COM

. . . queremos saber todos os arquivos que tenham extensão "COM" e que estejam no disco **default**. Se chamarmos nosso programa de PDIR, por exemplo, então o comando

PDIR *.COM

deverá listar os mesmos arquivos.

Esta tarefa, felizmente, é bastante facilitada pelo fato do CCP montar automaticamente o FCB do arquivo dado como parâmetro. Mais ainda. O CCP faz a substituição do caráter "*" por uma seqüência de "?"s correspondente – isto é importante pois as funções 17 e 18 do BDOS usadas no programa reconhecem as especificações ambíguas de arquivos constituídas de "?"s mas não aceitam as constituídas por "*"s.

Programa para mostrar o diretório de um disco

```
0000 =          BOOT    EQU     0           ; Endereço de reinicialização do sistema
0005 =          BDOS    EQU     5           ; Endereço para acessar o BDOS
005C =          FCB1    EQU     5CH         ; Endereço onde o CCP monta o FCB da linha de comando
0080 =          DMA     EQU     80H         ; Endereço do buffer de DMA
0009 =          IMPRIME EQU     9H          ; Função do BDOS para imprimir um buffer
0011 =          PROC1   EQU     11H         ; Função do BDOS para procurar o 1º arquivo
0012 =          PROCP   EQU     12H         ; Função do BDOS para procurar o próximo arquivo
0100            ORG     100H                ; Início do TPA

                Verifico se foi especificado um nome de arquivo

0100 3A5D00     LDA     FCB1 + 1    ; A → Primeiro caráter do arquivo
0103 FE20       CPI     "           ; Vejo se e branco
0105 C21C01     JNZ     PROCURA     ; Não: Vou direto para a procura
0108 3A6500     LDA     FCB1 + 9    ; A → Primeiro caráter da extensão
010B FE20       CPI     "           ; Vejo se e branco
010D C21C01     JNZ     PROCURA     ; Não: Vou direto para a procura

                Se não foi especificado arquivo, encho FCB1 com "?"s

0110 215D00     LXI     H, FCB1 + 1 ; HL → Endereço do FCB da linha de comando + 1
0113 060B       MUI     B, 11       ; Vou colocar 11 "?"s

                POE $ INTERROGA:

0115 363F       MUI     M, "?"      ; Ponho um "?"
0117 23         INX     H           ; Incremento o endereço
0118 05         DCR     B           ; Decremento o contador de "?"s
0119 021501     JNZ     Poe$Interroga ; Se diferente de 0, ponho outro

                Procuro o primeiro arquivo compatível com FCB1

                PROCURA

011C 115C00     LXI     D, FCB1     ; DE → endereço do FCB1
011F 0E11       MUI     C. PROC1    ; C → Func. BDOS para procurar o 1º arquivo
0121 CD0500     CALL    BDOS        ; Chamo o BDOS
0124 3C         INR     A           ; Vejo se A = 0FFH
0125 C2401      JNZ     Imprime$arq. ; Não: Imprimo o nome do arquivo

        Se não há arquivos compatíveis ; envio mensagem indicando isso e termino

0128 113301     LXI     D. SEM$ARQ  ; DE → Endereço da mensagem
012B 0E09       MUI     C. IMPRIME  ; C → Func. BDOS para imprimir buffer
012D CD0500     CALL    BDOS        ; Chamo o BDOS
0130 C30000     JMP     BOOT        ; Termino o processamento

0133 4E616F2068SEM$ARQ  DB          ; Não há arquivo$

                Imprime o nome do arquivo

                IMPRIME$ARQ

0142 3D         DCR     A           ; Devolvo A ao seu valor original
0143 87         ADD     A           ; Somo A com ele mesmo 5 vezes para
0144 87         ADD     A           ; obter o deslocamento dentro
```

O único probleminha que o programa precisa resolver é a possibilidade do usuário não declarar nenhum parâmetro. Neste caso, o que desejamos é obter a listagem de todos os arquivos, isto é, seria equivalente a especificar *.* como parâmetro. Em tal caso, no entanto, o CCP monta um FCB composto apenas de brancos, e não de "?"s como seria desejável – esta tarefa o próprio programa tem que efetuar. O que o programa faz, então, é verificar se o FCB montado pelo CCP tem um branco no primeiro caráter tanto do nome quanto da extensão – neste caso, substitui o FCB por uma seqüência de "?"s.

Já vimos que as funções 17 e 18 do BDOS retornam um código no acumulador indicando (1) se existe algum arquivo com um nome compatível com o especificado (2) onde procurar, dentro do buffer de DMA, o FCB do arquivo encontrado pela função.

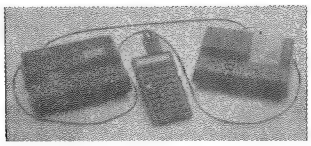
O que o programa faz, a seguir, é simplesmente chamar a função 17 do BDOS para localizar o primeiro arquivo (se não houver, uma mensagem indicativa é mostrada na tela, usando a função 9 do BDOS), e prosseguir num laço de repetição até o último arquivo da série. A cada iteração colocamos um ponto entre o nome e a extensão do arquivo, acrescentamos um caráter de tabulação (ASCII 9) para separar os nomes sucessivos na relação um caráter $, para indicar o fim do buffer para a função 9 do BDOS.

Depois disso, basta terminar a execução do programa reinicializando o sistema através de um "CALL 0".

Acredito que o programa em si está razoavelmente documentado de forma que, com o auxílio das explicações acima, o leitor pode compreendê-lo com relativa facilidade.

```
0145  87          ADD   A          ; do buffer de DMA onde o BDOS
0146  87          ADD   A          ; colocou o nome do arquivo
0147  87          ADD   A          ;
0148  218000      LXI   H, DMA     ; HL → Endereço do buffer de DMA
0148  5F          MOV   E, A       ; E → Deslocamento
014C  1600        MVI   D, 0       ; D →0
014E  19          DAD   D          ; HL tem agora o endereço do FCB
                                     do arquivo
014F  23          INX   H          ; Somo 1 p/ ter o end. do 1º caráter
                                     do nome
0150  EB          XCHG             ; Salvo em DE

                  Coloco um ponto entre o nome a extensão

0151  062E        MUI   B, '.'     ; Carácter a ser inserido
0153  0E03        MUI   C, 3       ; Número de caracteres a deslocar
0155  210800      LXI   H, 8       ; Número de caracteres do nome
0158  19          DAD   D          ; HL tem agora o end. do 1º caráter
                                     da extensão

                  DESLOC.

0159  7E          MOV   A,M        ; Salvo o caráter da extensão em A
015A  70          MOV   M, B       ; Substituo pelo que está em B
015B  47          MOV   B, A       ; Ponho o próximo caráter em B
015C  23          INX   H          ; Incremento HL para ter o end. do
                                     próximo car.
015D  0D          DCR   C          ; Decremento o contador de caracte-
                                     res em C
015E  C25901      JNZ   DESLOC     ; Se não chegou a 0, repito
0161  70          MOV   M, B;      ; Coloco o último caráter da extensão
0162  23          INX   H          ; Próximo caráter
0163  3609        MVI   M, 9       ; Acrescento um caráter de tabulação
0165  23          INX   H          ; Próximo caráter
0166  3624        MVI   M, S       ; Acrescento o "$" para marcar o
                                     fim do buffer

0168  0E09        MVI   C, IMPRIME ; C → Func. BDOS para imprimir
                                     buffer
016A  CD0500      CALL  BDOS       ; Chamo o BDOS para imprimir o
                                     nome do arquivo

                  Procuro o próximo arquivo

016D  115C00      LXI   D, FCB1    ; DE →Endereço de FCB1
0170  0E12        MVI   C, PROCP   ; C → Func. BDOS para procurar o
                                     próx. arquivo
0172  CD0500      CALL  BDOS       ; Chamo o BDOS
0175  3C          INR   A          ; Verifico se A = 0FFH
0176  C24201      JNZ   Imprime$arq ; Não: Imprimo o nome o arquivo
0179  C30000      JMP   BOOT       ; Sim: Término
017C              END              ; Fim do programa
```


QUANDO O MAIS IMPORTANTE É A SOLUÇÃO CERTA! HEWLETT PACKARD NA J. HEGER.



Sistema HPIL

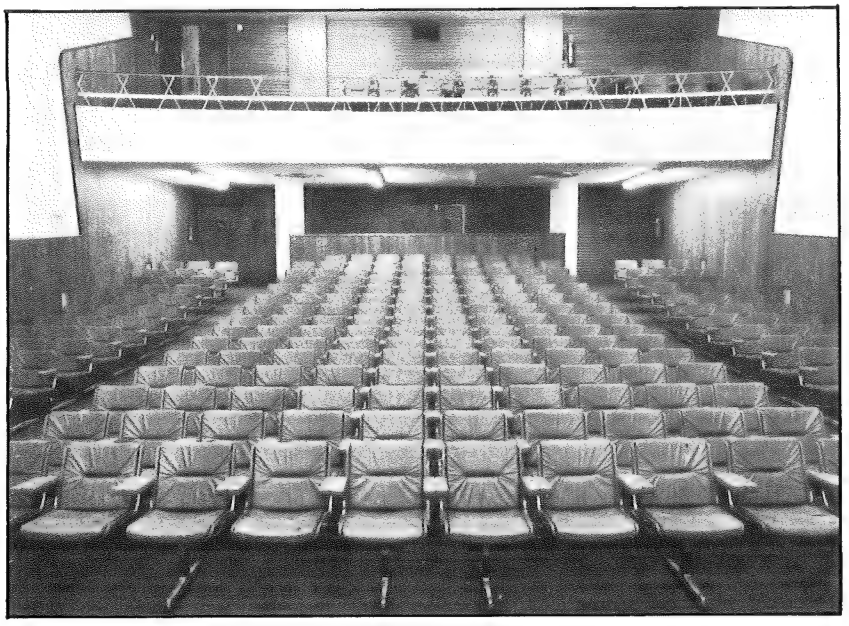
- Toda linha de calculadoras HP, em até 5 pagtos, ou 10% de desc. em 3 vezes, ou 15% de desc. à vista, sob os preços de tabela.
- Contrato de Manutenção e Extensão de Garantia - Opcional.
- Posto de Recebimento p/Assistência Técnica.
- Cursos Específicos p/calculadoras HP.
- Reembolso Varig e Serca p/todo o Brasil.
- Computador Técnico Científico HP-85 B.
- Consulte-nos!

J.Heger & Cia.
REVENDEDOR AUTORIZADO

Av. Mosci, 155 - Moema
Tel. (011) 531.7611
Telex (011) 23214
(ao lado do Shopping C. Ibirapuera)

HEWLETT PACKARD

impel INDÚSTRIA DE MÓVEIS PARA ESCRITÓRIO LTDA.



A INFORMÁTICA E A INDÚSTRIA DO MOBILIÁRIO

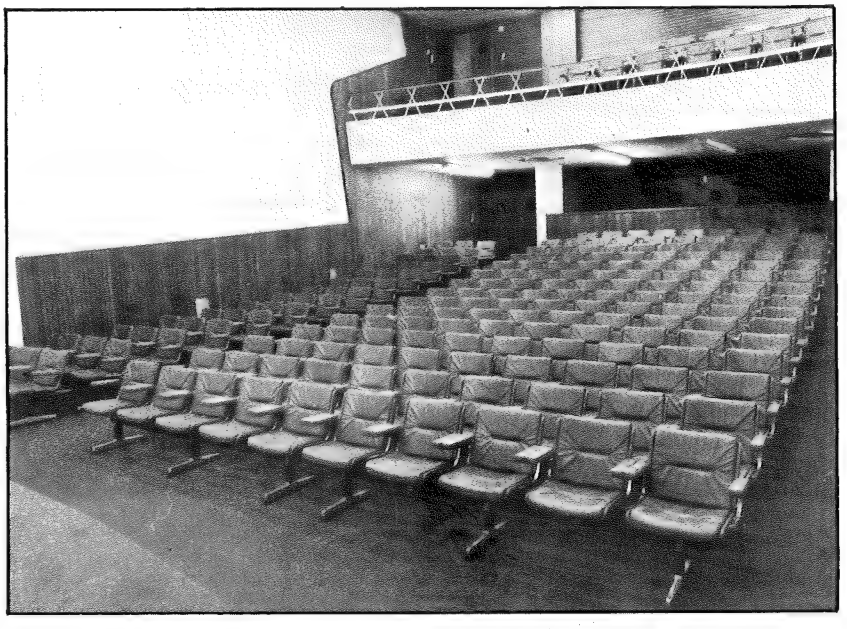
Por trás do mundo fascinante dos micros e sua extensa família, cujos membros têm as denominações mais extravagantes pra nós leigos, existe considerável gama de Indústrias que direta e indiretamente apoiam o setor da Informática, fornecendo peças, componentes ou produtos auxiliares.

É o caso da Indústria do mobiliário que vem dando sua contribuição, modernizando seus produtos, muitas vezes tradicionais adequando-os às necessidades de uso dos equipamentos de Informática.

A IMPEL INDÚSTRIA DE MÓVEIS PARA ESCRITÓRIO LTDA, Empresa com larga experiência e tradição no setor mobiliário, vem acompanhando com especial interesse o desenvolvimento do setor de Informática, tendo introduzido em sua linha de produtos, o mobiliário adequado e exigido pelo usuário desse novo equipamento de escritório.

Utilizando tecnologia própria e inteiramente nacional, a IMPEL está criando e lançará até o final deste ano, nova linha de cadeiras e poltronas com design totalmente voltado para as exigências de conforto, levando em conta as necessidades ergométricas e de preservação de saúde do usuário-operador, combinados com a moderna tendência de decoração do ambiente de trabalho.

É a nossa contribuição para tornar o mundo fascinante da Informação mais bonito e confortável, cumprimento e dever de apoio do setor moveleiro a esse importante segmento da moderna administração.



impel
IND. DE MÓVEIS PARA ESCRITÓRIO LTDA.

LOJA FARIA LIMA - SHOW ROOM
Av. Brig. Faria Lima, 1.480
Fones: 813.8311
São Paulo - SP.

LOJA SÃO PAULO
Viaduto Jacareí, 159
Fones: 35.8914 - 34.9346
São Paulo - SP.

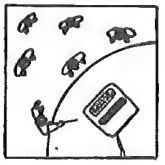
LOJA RIO
Rua Santa Luzia, 255-B
Fones: 220.4473 - 220.3573
Rio de Janeiro - RJ.

FÁBRICA -
Rua Consul Orestes Correa, 41
Fones: 209.9099 - 209.9268
209.9467 - Guarulhos - SP.

INFORMAÇÕES ÚTEIS PARA QUEM TEM, OU NÃO TEM O MICRO.

FILCRES

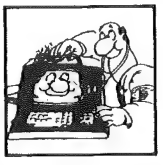
A Filcres faz de sua empresa o seu Show Room



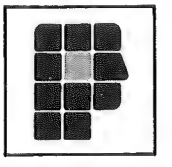
! Especialistas em

microcomputadores levam até você toda sua estrutura de Marketing. Conheça os CP300 e CP500 aliados ao alto desempenho da Impressora P500 e na configuração exata do seu problema.

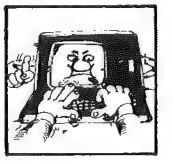
A Filcres oferece aos seus usuários assistência técnica



autorizada Prológica



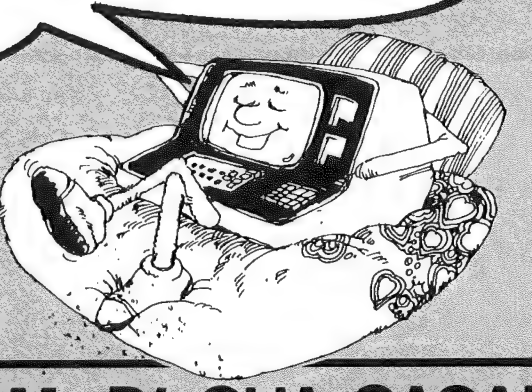
, completa biblioteca de software, diversificada linha de suprimentos, além de treinamento gratuito de operação



e linguagem Basic

.

Venha até aqui, ou ligue que iremos até você!



filcres

FILCRES ELETRÔNICA ATACADISTA LTDA.
Rua Aurora, 165 – CEP 01209 – São Paulo – SP
Tels.: 223-1446 – 220-5794 – 222-3458
PBX: 223-7388

LEVE NOSSO SHOW ROOM P/ SUA CASA!

O MELHOR EM HARDWARE

1 - MICROPROCESSADOR BIT SLICED: UMA OPÇÃO PARA 32 BITS

César da Costa

(Revistas Interface Vol. 1, nº 5, págs. 13 a 15; Vol. 1, nº 6, págs. 15 a 20; Vol. 1, nº 7, págs. 16 a 22; Vol. 1, nº 8, págs. 16 a 20; Vol. 1, nº 9, págs. 16 a 20)

1.1 – Arquitetura do Microprocessador Bit Sliced
1.2 – Microprocessador Bipolar
1.3 – Aplicações
1.4 – Visão Geral da arquitetura
1.5 – Seção de processamento
1.6 – Seção de controle
1.7 – Operações lógico-aritméticas
1.8 – Características da microprogramação
1.9 – Controle microprogramado X Controle por hardware
1.10 – Funções microprogramadas
1.11 – Formatação e codificação de microinstruções
1.12 – Microprogramação vertical e horizontal
1.13 – Aspectos de microprogramação
1.14 – Baixo nível
1.15 – Alto nível
1.16 – Implementação usando componentes MSI
1.17 – Implementação de seqüenciadores com FPLA (FIELD PROGRAMMABLE LOGIC ARRAYS)
1.18 – Seqüenciadores comerciais de microprogramas LSI
1.19 – Considerações gerais
1.20 – Considerações de projetos de sistemas de controle
1.21 – Suporte para o projeto
1.22 – Quando um sistema microprogramável BIT SLICE faz sentido

MICROPROCESSADOR BIT SLICED: UMA OPÇÃO PARA 32 BITS

Com o alto avanço da tecnologia bipolar e a flexibilidade de microprogramação pelo usuário, os microprocessadores bit sliced encontraram espaço em aplicações críticas, sendo uma opção tecnológica para os superminicomputadores que estão em pauta na atual tecnologia brasileira.

No início da década passada, o surgimento dos microprocessadores deram uma nova dimensão aos projetistas de equipamentos digitais. Esta tecnologia ampliou o **spectrum** de aplicações, pois os novos dispositivos LSI eram mais baratos, menores, requeriam menos energia e permitiam um projeto mais versátil.

Um microprocessador é basicamente um circuito integrado que corresponde à CPU de um computador digital. Este dispositivo, associado a uma memória de acesso randomico (RAM), para o armazenamento de dados e resultados intermediários; alguma memória somente de leitura (ROM), para armazenar as instruções; uma fonte de alimentação; e os circuitos de interfaces para o I/O, fazem um microcomputador. Este microcomputador satisfaz a todos os requisitos mínimos de um computador, ou seja, pode receber dados e transmiti-los; contém uma unidade de processamento de dados e controles para execução das operações; contém um mínimo de memória (RAM, ROM, EPROM. etc) para o armazenamento de dados e instruções; é programável pelo usuário e bastante rápido, podendo executar uma instrução em menos de 10 microsegundos.

Os microprocessadores convencionais são fabricados usando-se a tecnologia CMOS; a implementação mais comum desses chips coloca ambas as funções, a de processamento de dados e a de controle, num único chip.

Os microprocessadores CMOS tornaram-se populares nos últimos anos, sendo utilizados em um número continuamente ascendente de aplicações. Devido ao tamanho extremamente pequeno e ao baixo custo, estes dispositivos revolucionaram os sistemas digitais, substituindo o hardware convencional e eliminando uma série de chips SSI e MSI; ainda, reduzindo o consumo de energia e aumentando a confiabilidade do sistema. A maior desvantagem dos dispositivos CMOS é a velocidade relativamente baixa, quando comparada com os dispositivos bipolares.

MICROPROCESSADOR BIPOLAR

Os microprocessadores **bit sliced** fabricados com a tecnologia bipolar, se comparados com os microprocessadores **CMOS**, fornecem um sistema digital com muito mais versatilidade e uma velocidade de processamento 10 (dez) vezes maior. Estes dispositivos microprogramáveis permitem ao projetista de sistema superar a maioria das deficiências que possam surgir no projeto, mesmo após a sua implementação.

Os dispositivos bipolares bit sliced diferem dos microprocessadores CMOS principalmente na filosofia da arquitetura de suas CPU's. Para os microprocessadores CMOS a função de processamento de dados e a função de controle são ambas feitas por hardware no mesmo chip) por exemplo, o circuito de decodificação das instruções). Por esta razão, os microprocessadores CMOS são também chamados de microprocessadores de instruções fixas.

Nos microprocessadores bipolares as duas funções da CPU, processamento e controle, são realizadas em chips separados. Enquanto os microprocessadores CMOS têm pré-definido e imutável o comprimento da palavra (word length), a arquitetura e o conjunto de instruções, os microprocessadores **bit sliced** podem ser configurados por microprogramação, fornecendo uma ampla variedade de arquiteturas com vários comprimentos de palavras e capacidade de conjuntos de instruções.

O projetista pode definir o conjunto de instruções do sistema por meio de um programa (microprograma) armazenado numa EPROM. Cada instrução que o usuário vê, está programada neste dispositivo. Qualquer pessoa pode otimizar a arquitetura do sistema e a capacidade de processamento para os requisitos da aplicação. Ao mesmo tempo, os sistemas finais tornam-se muito mais flexíveis, pois os melhoramentos, implementações ou alterações em vista na arquitetura do sistema, podem ser alcançadas trocando-se algumas microinstruções ou substituindo por inteiro a memória de controle EPROM que as armazena.

Estas vantagens, porém, não são independentes. Para alcançá-las é necessário um grande investimento no suporte de sofware do sistema.O engenheiro projetista deve ter um bom conhecimento de hardware e software, um claro entendimento sobre as diversas arquiteturas do sistema e a habilidade da microprogramação. Trabalhar com microprocessadores **bit sliced** é mais difícil e consome mais tempo do que trabalhar com microprocessadores CMOS. As microinstruções são mais complicadas que as instruções em linguagem de máquina (macroinstruções), logo, a microcodificação leva mais tempo. Os recursos de microprogramação não são tão ricos como os de macroprogramação, porém, a geração de microinstruções pelo usuário é uma facilidade que aumenta o desempenho dos sistemas com **bit sliced.**

APLICAÇÕES

A microprogramação e os microprocessadores **bit sliced** são usados freqüentemente em projetos de sistemas, devido ao fato que, para certas aplicações, elas podem representar o caminho mais apropriado para substituir o hardware atual, fornecendo a única maneira prática de alcançar as funções especiais desejadas.

Algumas notáveis áreas de aplicação incluem processamento de sinal (filtragem digital), sistemas on-line, comunicação de dados, controle de processo e controladores de alta velocidade para unidades, tais como discos, vídeos, displays gráficos e emulação executando algumas ou todas as instruções de outros computadores existentes. Outra grande área na qual estes dispositivos têm tremendo impacto é a implementação de algoritmos que são suscetíveis a realizações paralelas de hardware. E por que não os superminicomputadores nacionais?

VISÃO GERAL DA ARQUITETURA

Tendo em vista que o software de apoio à disposição é demasiadamente pobre, o engenheiro projetista precisa ter um conhecimento detalhado da organização da arquitetura do sistema. A maior parte do esforço do projetista, além das instruções programadas, deve ser o controle direto dos elementos específicos do hardware, que levam às restrições de temporização, ou seja, problemas de propagação.

A maioria dos sistemas **bit sliced** microprogramáveis tem seu diagrama de bloco básico ilustrado, na figura 1. Ao contrário dos microprocessadores CMOS, os microprocessadores **bit sliced** usam a tecnologia **Schottky** bipolar. Devido à complexidade funcional interna, às limitações comuns de implementação e ao número de pinos e tamanho de chip possível com a tecnologia bipolar, o microprocessador **bit sliced** tem que ser implementado numa base multichip. A seção de controle e a de processamento da CPU estão localizadas em chips diferentes e são construídas de forma diversa das equivalentes CMOS.

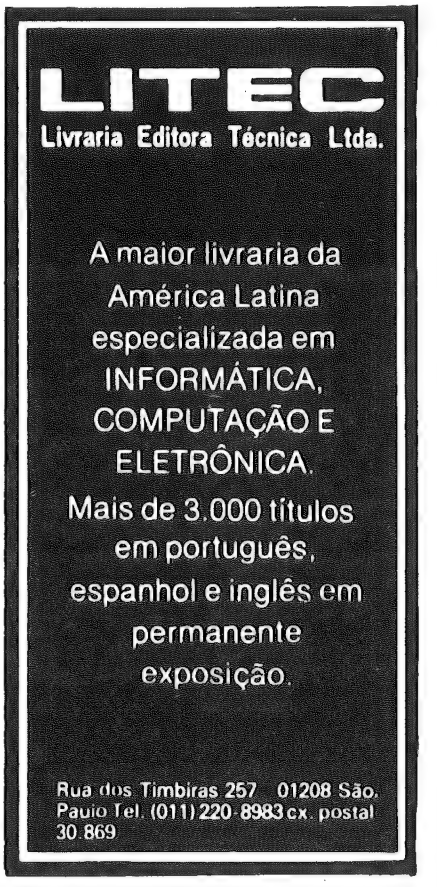
SEÇÃO DE CONTROLE

Esta seção representa o projeto mais complexo e dá à máquina sua "personalidade" ou conjunto de instruções. Consiste tipicamente da memória microprogramada, também chamada memória de controle ou controle de armazenamento; um contador microprogramado, também chamado unidade de controle microprogramada ou controlador; e alguns circuitos restantes, tais como lógica de seleção e registradores **Pipeline**; cada um sempre implementado num chip LSI separado. A memória microprogramada contém as microinstruções que especificam os passos através dos quais a máquina seqüencia e controla a operação paralela da RALU (Registro da Unidade Lógica e Aritmética). O contador de microprograma fornece a macroinstrução de decodificação lógica e determina o **próximo microendereço**, gerando o esquema para seqüenciação dos microprogramas.

A memória de microprogramação (ROM ou EPROM) expande-se para conter um grande número de palavras de instruções microprogramadas. Cascateando-se uma quantidade de EPROM's similares, a largura da microinstrução é expandida, mas não necessariamente o número de funções por bit. Mais detalhes da arquitetura interna da seção de controle de operação será dada posteriormente.

SEÇÃO DE PROCESSAMENTO

Nesta seção são executadas todas as operações lógicas e aritméticas. É composta de chips funcionalmente equivalentes, chamados **bit slices** ou **RALU's**. Cada RALU contém uma quantidade de registradores de propósito geral e específico, um acumulador, uma unidade lógica aritmética e flags de status. Atualmente, a maioria dos microprocessadores bipolares é construída usando-se 8 bits ou 16 bits slices. Estes slices operam em paralelo e podem ser cascateados a qualquer extensão que seja múltiplo do slice básico. Desta maneira, podemos construir microprocessadores que possam suportar extensões de palavras não-convencionais, por exemplo, 24, 32 e 48 bits.



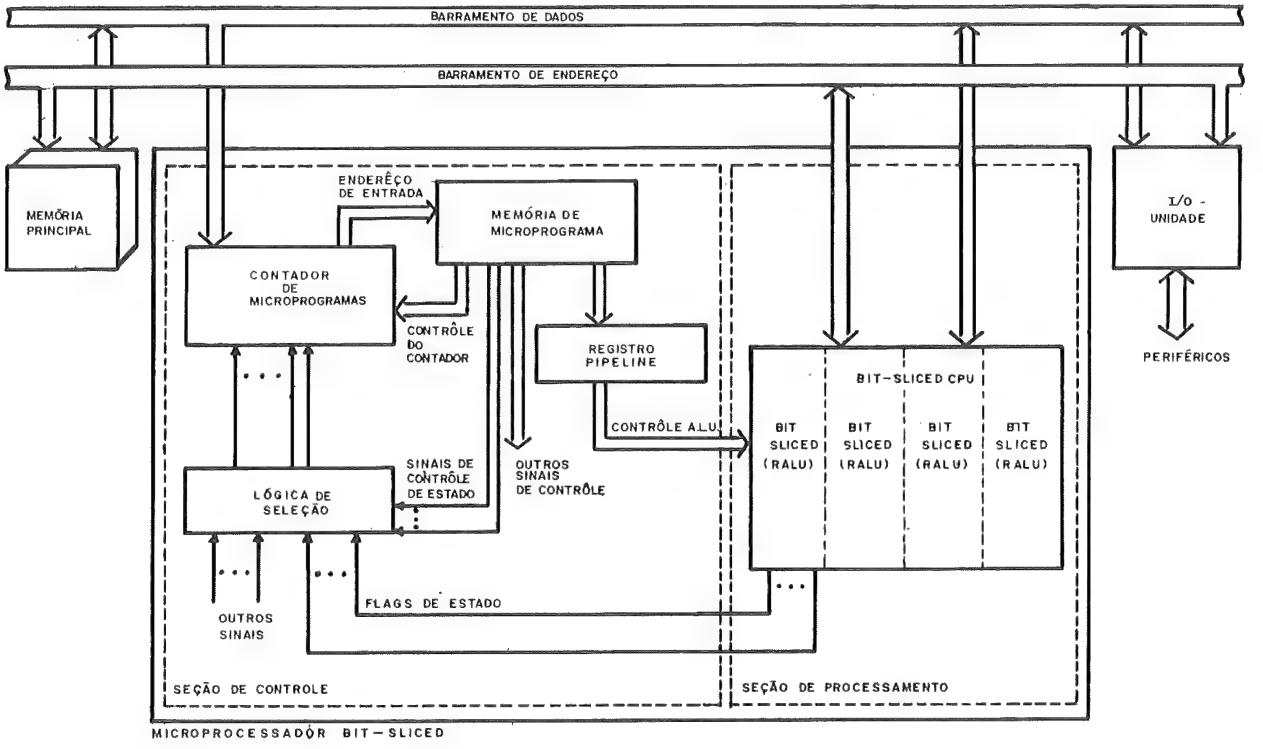


FIGURA 1 — MICROCOMPUTADOR BIT SLICED MICROPROGRAMÁVEL

sistema

ASSISTÊNCIA TÉCNICA

☐ Assistência a Micros Nacionais
Todas as marcas e modelos Importados:
Sinciair - Trs-80 - Apple-IBM

Outras marcas poderão ser atendidas

Assistência Técnica autorizada
PROLOGICA microcomputadores

Av. Presidente Vargas, 542
Sala 2111 - Tel.: 253-0645
Rio de Janeiro

DATAMICRO

VENDA DE MICROCOMPUTADORES
TK 83, 85, & 2000 COLOR
CP 300, 500 & 600
COLOR 64 (EXT. BASIC)

SUPRIMENTOS
Disquetes, fitas, form. contínuos

CONSULTORIA DE SISTEMAS
Diagnóstico e apoio à decisão

CURSOS E TREINAMENTO
Introdução aos microcomputadores
Linguagem Basic
Aplicação dos micros na Engenharia
Microcomputadores para crianças

INSCRIÇÕES ABERTAS

Livros e revistas especializados

Visc. de Pirajá, 547 Sobreloja 211
Cep. 22410 Ipanema Rio RJ
Tel.: (021) 274-1042
DESPACHAMOS PARA TODO O BRASIL

O FUTURO NÃO SE ARRASTA. VOA...!
Fique ligado!

Micro computadores
- MICRODIGITAL — PROLÓGICA
- TK 83 TK 85 TK 2.000
- CP 200 CP 300 CP 500
- PROGRAMAS COMERCIAIS E JOGOS
- SUPRIMENTOS

Cursos
- INTRODUÇÃO À MICROCOMPUTADORES BASIC I E II
- AULAS PRÁTICAS E TEÓRICAS
- 2 A 3 ALUNOS POR MICRO
- TURMAS DE 12 ALUNOS
- APOSTILAS E CERTIFICADOS
- TURMAS ESPECIAIS PARA EMPRESAS

telecomunic
engenharia e comércio ltda

DEPTO. DE INFORMÁTICA
AV. BRIG. FARIA LIMA, 2178 • TEL.: 813-3338
PINHEIROS — SÃO PAULO - SP
ESTACIONAMENTO ZONA AZUL

OPERAÇÃO DE CONTROLE

Na configuração mostrada na figura 1 o microprocessador conduz as macroinstruções da memória principal do sistema sob a direção das microinstruções lidas da sua memória microprogramada. O código de operação da macroinstrução é interpretado pelo contador de microprograma, isto é, mapeado num endereço de memória do microprograma e depois executado como uma série de microinstruções.

O conteúdo relativo à operação da macroinstrução é dirigido às RALU's, para ser usado tanto no processamento como na manipulação dos endereços da memória principal. Deste modo, enquanto a memória principal do sistema contém macroprogramas (programas de aplicação), a memória microprogramada da seção de controle contém microprogramas que definem o microprocessador.

A memória microprogramada, vista também como uma máquina de estado programável, gera pulsos de controle temporizados para controlar o restante do sistema. Os estados das suas linhas de controle de saída são representados pela microinstruções armazenadas na memória microprogramada. O contador é a unidade que determina as entradas da memória microprogramada, baseado nos vários bits de estado da máquina e em alguma realimentação recebida da memória.

OPERAÇÕES LÓGICO-ARITMÉTICAS

A parte mais importante da seção de processamento é a ALU, que executa todas as operações lógicas e aritméticas. No passado, usava-se a técnica de implementação modular de ALU, que consistia no cascateamento de uma quantidade de somadores paralelos de 4 bits (74181), capazes de executarem funções lógicas e aritméticas no tamanho desejado. O uso do 74281 (ALU) representou um melhoramento, pois ele incluiu um acumulador interno e uma matriz de deslocamento. Atualmente, utiliza-se a distribuição vertical de CPU's bit sliced, cascateando-se RALU's padrão, cuja capacidade adicional reside no fato de que contém uma série de registradores internos muito úteis para armazenamento de dados temporários.

As RALU's comercialmente disponíveis também permitem detectar condições internas tais como overflow e zero; contém o barramento necessário à decodificação lógica para controle da ALU e os registros de transferência sob controle da instrução, todos os **slices** operam em paralelo. Adicionando-se mais **slices** RALU obtém-se a expansão do tamanho da palavra **(word length)** do sistema.

Em alguns casos os registros **PIPELINE** podem ser inseridos entre a memória microprogramada e as RALU's, a fim de melhorar o desempenho e a velocidade total do sistema. Esta implementação é conseguida superpondo a execução da microinstrução com a busca (FETCH) da próxima microinstrução.

Usando-se microprogramas, muitas macroinstruções complexas podem ser implementadas como seqüências de microinstruções relativamente primitivas. A lógica da macromáquina permanece relativamente simples e o usuário do sistema não precisa saber se o projetista implementou o microprocessador através de esquemas microprogramáveis. No entanto, o usuário tem a opção de entrar em contato com o projeto detalhado a fim de alterar a arquitetura da máquina para aumentar a sua capacidade e otimizá-la para outras aplicações.

A figura 2 mostra dois dos microprocessadores bit sliced microprogramados mais comumente usados, o INTEL 3000 e o AMD 2900.

CARACTERÍSTICAS DA MICROPROGRAMAÇÃO

A microprogramação, desde a sua iniciação por Wilkes em 1951 e a sua realização comercial nas séries IBM System/360 em 1964, tem se tornado cada vez mais popular. De acordo com Flynn, a microprogramação tem evoluído através de 3 fases distintas. A primeira fase usava a tecnologia matriz de diodo para a memória microprogramada e a escolha da próxima microinstrução a ser executada era influenciada por certos bits de condição. As matrizes de diodo, com um tempo de acesso abaixo de 0,5 microsegundos, comparadas com o tempo de ciclo de dez microsegun-

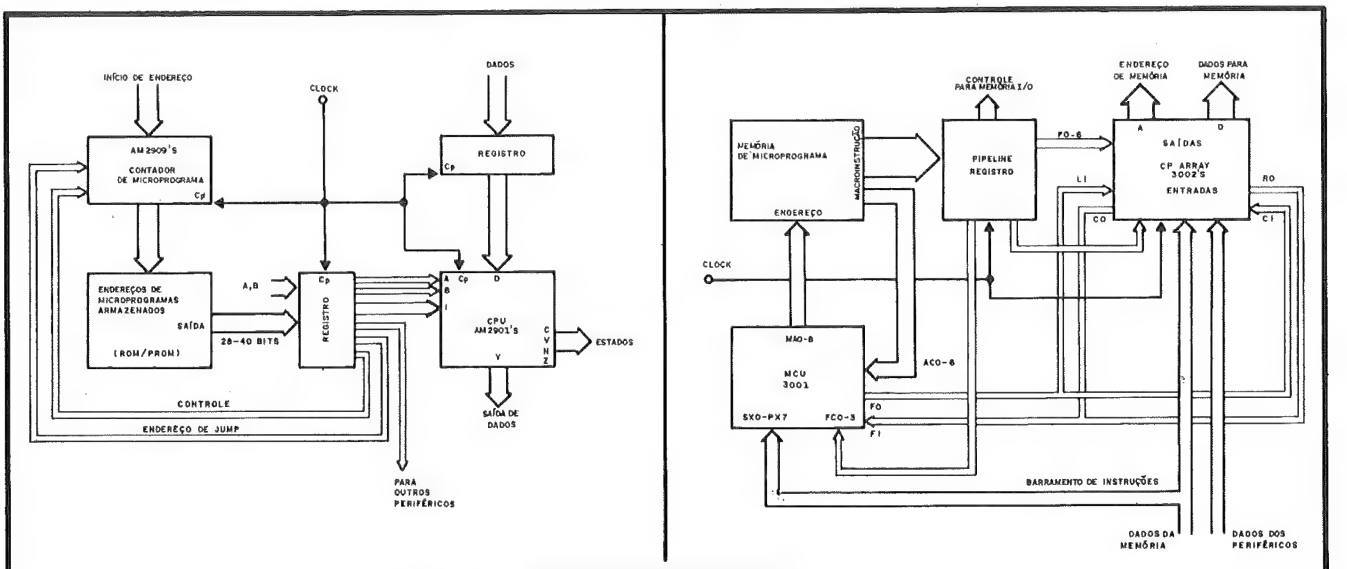


FIGURA 2 – MICROPROCESSADORES BIT SLICED MICROPROGRAMADOS:

dos da memória principal, foram rápidas o suficiente para não causar diminuição no desempenho.

O aparecimento dos dispositivos programáveis LSI conduziram à atual fase de microprogramação.

Com o aparecimento dos dispositivos programáveis LSI conduziram à atual fase de microprogramação.

Com o aparecimento das memórias de núcleo magnético rápidas, durante os anos 50, esta velocidade baixa de microarmazenamento começou a apresentar sérios problemas, pois as velocidades da memória prinicipal tinham agora caído para menos de 1 microsegundo, enquanto que a velocidade de armazenamento de ROM usada para a memória do microprograma variava entre 200-400 nanosegundos. Assim, durante esta segunda geração, a principal aplicação e uso da microprogramação parecia ser para estimular outras máquinas num único sistema **host**.

A 3ª geração da microprogramação começou no final dos anos 60, quando apareceram as rápidas memórias bipolares monolíticas de controle de leitura e escrita. Estas memórias contribuíram para o projeto de sistemas dinamicamente microprogramáveis. Nós agora tínhamos a habilidade para adequar a arquitetura do sistema à aplicação disponível, ou mudar a configuração de um sistema defeituoso, de modo que ele pudesse se tornar operacional passo a passo, possibilidade que era quase impossível com os sistemas tradicionais.

Em 1977, Rosin observou que a história passada indicava que a microprogramação tinha realmente sido usada de um modo equivalente à interpretação, isto é, para ocultar uma porção de erros de projeto de máquinas basicamente mal estruturadas.

FASE ATUAL

Com o aparecimento dos dispositivos LSI programáveis, tais como processadores bit sliced, registros **PIPELINE**, matrizes lógicas programáveis e PROM's, iniciou-se uma 4ª fase de microprogramação que começou por volta de 1974/75. Os microprogramas podiam agora operar agregados em **slices** em paralelo. Este paralelismo, juntamente com a capacidade de definir o comprimento de palavra, aumentou tremendamente a eficácia das velocidades de processamento e a flexibilidade do sistema. A microprogramabilidade do usuário e os microprocessadores bit sliced tornaram os conceitos arquiteturais mais populares nos últimos anos. Os projetistas de sistemas de hoje estão também interessados em usar técnicas de microprogramação em aplicações fora da CPU. As memórias bipolares LSI são econômicas e rápidas, tornando-se de uso prático para técnicas de microprogramação num vasto alcance de sistemas digitais complexos.

CONTROLE MICROPROGRAMADO X CONTROLE POR HARDWARE

A função de uma **seção de controle** é fornecer os sinais de controle utilizados pelo resto da máquina e determinar quais serão os próximos sinais de controle. Durante o ciclo de busca da operação da máquina, a seção de controle interpreta o código de operação da instrução que chegou da memória principal. Isso geralmente resulta numa mudança de estado do sistema e produz um estímulo que é enviado para as outras unidades da máquina. Na realidade, uma seqüência desses sinais de controle devem ser gerados, já que a execução de uma simples instrução causa mais de uma mudança de estado. Cada um desses grupos de sinais produzem uma operação básica específica (chamada microoperação) que é executada como uma transferência de informação entre dois registros ou como uma mudança dos conteúdos de um registro, ou ainda, como que carregando os resultados de saída de um registro. Os próximos sinais de controle podem depender ou não dos resultados da microoperação presente ou do estado de certos **flags** do sistema que são gerados. Assim, a interpretação e execução de uma macroinstrução envolve uma seqüência de operações básicas da máquina (microoperações), cada uma controlada por um grupo específico de sinais de controle ou microinstruções. Esses sinais de controle ativam independentemente pontos de controle que basicamente coordenam as comunicações entre ciclos de registros para registros. A operação funcional e o fluxo de dados internos da seção de processamento são geralmente considerados independentes do fato do controle ser por hardware ou microprogramado.

A figura 3 mostra os diagramas de bloco básicos tanto da seção de controle clássica por hardware como da seção de controle microprogramada.

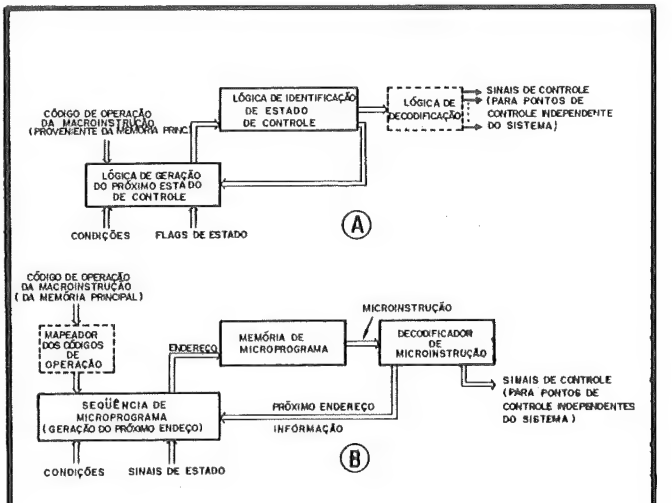


FIGURA 3 – SEÇÃO DE CONTROLE DE UM COMPUTADOR
(a) IMPLEMENTAÇÃO CLÁSSICA POR HARDWARE
(b) IMPLEMENTAÇÃO POR MICROPROGRAMA

Como foi mencionado anteriormente, a tarefa principal da seção de controle é a de seqüenciar as microoperações da máquina e gerar os sinais de controle para os pontos de controle do sistema. Esta tarefa tem duplo objetivo: primeiro, o estado de controle durante cada ciclo deve ser estabelecido, e segundo, o próximo estado de controle seqüencial correto deve ser determinado. Portanto, a seção de controle deve conter uma lógica para o armazenamento da informação, que identifica o estado de controle utilizado e uma lógica de decisão para a geração (escolha) do próximo estado de controle. A seção de controle por hardware é implementada pelo uso de combinações de portas, contadores e decodificadores, todos interconectados de um modo não-randômico. Esta seção de controle não é flexível, e tem dificuldades de projeto que conduzem a configurações não-estruturadas e bem complexas.

"Trabalhar com Microprocessadores Bit Sliced é mais difícil e consome mais tempo do que trabalhar com Microprocessadores CMOS"

"As vantagens dos Microprocessadores Bit Sliced não estão sozinhas, elas requerem um sério investimento em suporte de Software"

A principal distinção física entre a seção de controle por hardware, figura 3-A, e a microprogramada, figura 3-B, é que a primeira usa dispositivos MSI (registros) para armazenagem de informações de controle, enquanto a última usa uma forma de matriz de memória LSI organizada regularmente (ROM, PROM, PLA) chamada memória de micro-programa. Na implementação micro-programada o estado de controle da máquina é identificado pelo endereço da memória mocroprogramada, enquanto que os conteúdos desta locação (macroinstrução) fornecem a informação necessária para estabelecer as combinações do sinal de controle e escolher o próximo endereço. O contador de microprograma gera o próximo endereço da microinstrução, definindo o próximo estado de controle da máquina baseado na próxima informação de endereço recebida das entradas do código de operação, da macroinstrução, dos **flags** de estado, ou das entradas de condição recebidas do resto do sistema.

Vários grupos dessas microinstruções (microrotinas) são geralmente incorporados em um sistema complexo. O código de operação de uma instrução de máquina (macroinstrução), quando devidamente decodificada (mapeada) na seção de controle, aponta para uma microrotina apropriada na memória de microprograma. A execução da microrotina pelo hardware resulta no controle de todas as operações fundamentais, como transferência de dados e transformações elementares de dados. A mudança dessas microrotinas ou a substituição por outra memória microprogramada altera a complexidade funcional da máquina. Agora ela se comporta de um modo inteiramente novo, isto é, executa um grupo de macroinstruções completamente diferente, tem uma nova arquitetura, pode ser usada em aplicações específicas e pode emular outras máquinas.

O aspecto mais importante da seção de controle é a esturutura do contador de microprograma, já que esta estrutura é que determina as características gerais da seção de controle e os mecanismos de seqüenciação disponíveis usados para implementar a modularidade de microprogramas. Para garantir essa modularidade, a seção de controle deve permitir o uso das estruturas de controle básicas mostradas na figura 4. As várias maneiras de implementar o seqüenciamento de microinstrução são dadas no prosseguimento deste artigo. Uma classificação mais formal e detalhada das estruturas da seção de controle microprogramada pode ser encontrada nos livros de Marczynski e Tudruj, vide bibiografia.

FUNÇÕES MICROPROGRAMADAS

A fim de entender melhor os requisitos de projeto de uma seção de controle microprogramada apresentamos um exemplo que define as microoperações e os sinais de controle envolvidos na execução da macroinstrução "COMPARE". Essa instrução específica compara os conteúdos de dois registros A e B, e executa as seguintes ações:

Se A = B então A ← A*1/2 senão IC ← IC + 1

Em outras palavras, se os conteúdos dos dois registros são iguais, então a ação executada é fazer A = A* 1/2; do contrário, o contador de macroinstrução IC é aumentado de 1 e aponta para a próxima instrução do programa. Essas atividades são mostradas no diagrama em blocos, figura 5, e cada função é desempenhada num ciclo da seção de processamento. Aqui, assumimos que a máquina usa a notação de complemento de 2 para representar nºs negativos.

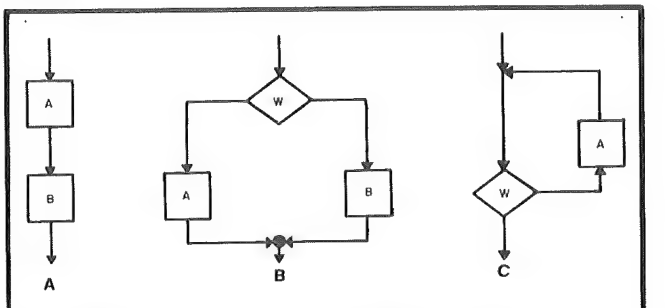


FIGURA 4 – ESTRUTURA DE CONTROLE BÁSICO IMPLEMENTADA POR MICROPROGRAMAS: (a) A então B; (b) Se W então A senão B; (c) Enquanto W faça A (onde A e B representam módulos de microprogramas)

Consideramos a seção de processamento como tendo o diagrama em blocos mostrado na figura 6. Esta seção de processamento simplificada contém dois registros A e B, uma ALU, um deslocador (**shift**) que pode deslocar o resultado para a direita ou esquerda e uma lógica de detecção de zero, que emite um **flag** quando todos os bits do barramento de dados são zero. O IC contém o endereço da ma-

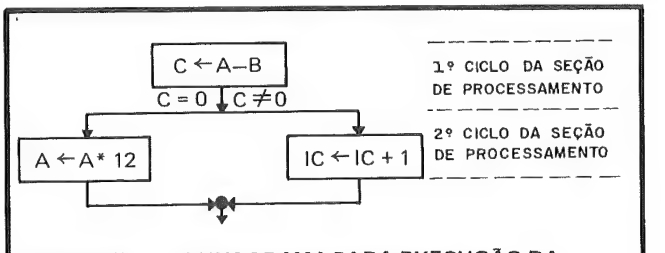


FIGURA 5 – FLUXOGRAMA PARA EXECUÇÃO DA INSTRUÇÃO "COMPARE"

"A seção de controle microprogramada dá ao Microprocessador Bit Sliced a sua personalidade"

croinstrução executada atualmente ("COMPARE"), e o registro de estado contém um bit que lembra o estado da lógica de detecção de zero. Finalmente, desde que as entradas para a ALU estejam na notação **complemento a um**, um bit **carry in** é necessário para convertê-las em **complemento a dois**, que é a notação requerida pelo sistema.

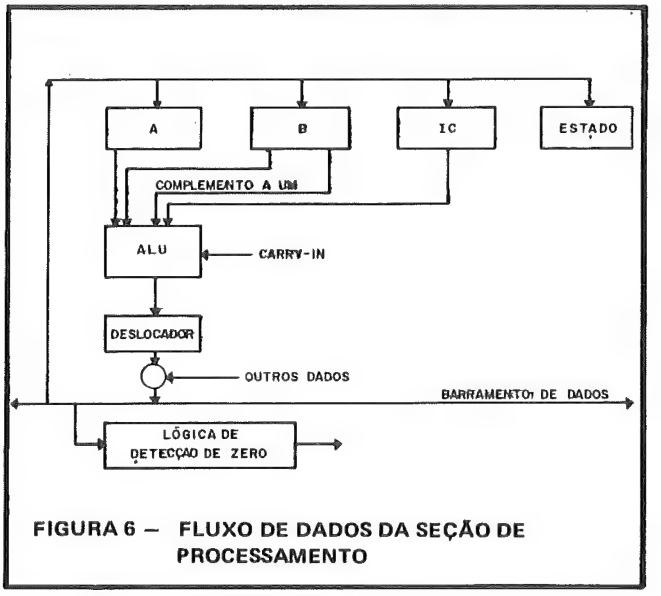


FIGURA 6 — FLUXO DE DADOS DA SEÇÃO DE PROCESSAMENTO

Tendo definido a estrutura da seção de processamento devemos agora definir o formato da microinstrução usado. Os sinais de controle apropriados também são necessários para executar todas as microoperações, selecionar os registros apropriados, transferir os dados entre eles, ou direcionar o resultado da ALU para um destino específico. Para o nosso exemplo simplificado de execução da macroinstrução "COMPARE", os seguintes sinais de controle são necessários:

- **Sinal A**

É usado para atuar como porta entre o conteúdo do registro A e a ALU. Se A = 0, então, A não serve como porta para a ALU, Se A = 1, então, A é porta para a ALU.

- **Sinais A e C**

As combinações destes dois controles de sinal determinam o caminho que o registro B e o IC são ligados à ALU, como segue:

00 : nem B nem C são ligados à ALU.
01 : o complemento B é ligado à ALU.
10 : o próprio B é ligado à ALU.
11 : o IC é ligado à ALU.

- **Sinal D**

É usado para controlar a ação de soma desempenhada sobre as entradas da ALU. Se D = 0, a soma é executada sem nenhum carry-in; se D = 1, a soma é executada com carry in.

- **Sinais E, F**

Estes dois controles de sinal especificam as ações de mudança, como segue:

00 : nenhum deslocamento no resultado (a saída da ALU é ligada diretamente ao barramento de dados)
01 : a saída da ALU é ligada ao barramento de dados, deslocamento de um à direita.
10 : a saída da ALU é ligada ao barramento de dados, deslocamento de um à esquerda.
11 : "outros dados" são ligados ao barramento de dados.

- **Sinal G, H**

Estes dois sinais de controle especificam o destino para o qual a informação no barramento de dados deve ser dirigida:

00 : nenhum destino
01 : a informação no barramento de dados é direcionada ao registro A.
10 : a informação no barramento de dados é direcionada ao registro B.
11 : a informação no barramento de dados é direcionada para o IC.

A fim de executar as três microoperações necessárias à execução da instrução "COMPARE", a seção de processamento deve receber os três conjuntos de sinais de controle de A até H (assumimos que cada conjunto é usado para controlar uma execução de microoperação):

	Sinais de Controle							
MICROOPERAÇÃO	A	B	C	D	E	F	G	H
C ← A – B	1	0	1	1	0	0	0	0
A ← A*1/2	1	0	0	0	0	1	0	1
IC ← IC + 1	0	1	1	1	0	0	1	1

Por exemplo, observamos que no caso da primeira microoperação, C ← A – B, os sinais de controle indicados servem de porta para a ALU. O registro A, o complemento a um do registro B e o bit carry-in especificam a ação de soma. Efetivamente, o conteúdo do registro B é subtraído do conteúdo do registro A. O resultado desta subtração é ligado ao barramento de dados (sem nenhum destino posterior) e simplesmente ajusta o bit de saída da lógica de detecção zero, se A = B.

Até agora definimos 8 bits das três microinstruções necessárias para emitir os sinais de controle requeridos. As situações situações reais necessitarão de muito mais bits de controle, assim como será necessário implementar microoperações adicionais, meios para executar macroinstruções de busca e intepretação, e outras funções.

No nosso exemplo, a execução da segunda ou terceira microoperação é determinada baseando-se no resultado da primeira microoperação. Especificamente, o sinal fora da lógica de detecção de zero, figura 7, deve ser lembrado no registro de estado, e este examinado na próxima seção de processamento para determinar qual a segunda ou terceira microoperação a ser executada em seguida. Um meio fácil de fornecer tal capacidade de ramificação condicional é aumentar a extensão da microinstrução para incluir uma condição extra (I, J), um controle de registro de estado (K), e um próximo endereço (Z) de 6 bits.

SACCO
computer store

Computadores

Linha IBM - PC

PC 2001 - MICROTEC - LINK 727

Redes Locais;
Comunicação Remoto/Local com IBM
Emulação de Terminal IBM
Comunicação IBM/Apple
Winchester; Software Especiais

Linha APPLE

CRAFT II plus - UNITRON AP II
Comunicação com Minis.

Impressoras

GRAFIX - ISA - OLIVETTI
ELEBRA - ELGIN - GLOBUS

Monitores de Vídeo

INSTRUM - VÍDEOCOMPO

Consultoria

Cursos

Software

Aplicativos
Processamento de Textos
Banco de Dados
Planílhas Eletrônicas

Específicos
Administrativos - Comerciais
Engenharia - Gráficos

Al. Gabriel Monteiro da Silva, 1229
São Paulo - Tel.: (011) 852-0799

CIÊNCIA MODERNA COMPUTAÇÃO

A única especializada em livros e revistas para microcomputadores

Você que tem apple não pode deixar de comprar.

PREÇO Cr$ 7.500

20 jogos inteligentes em applesoft

anita palmer

- Desafiantes
- Divertidos
- Educativos

20 JOGOS INTELIGENTES EM APPLESOFT

PARA TODA A LINHA APPLE

CIENCIA MODERNA COMPUTAÇAO LTDA

Visite-nos ou peça uma lista do nosso estoque específica para seu micro.

Atendemos tambem por Reemb Postal e Varig

Av Rio Branco, 156 — loja 127 — subsolo
Rio de Janeiro — Tels 262-5723/240-9327

SOFT MICRO'S

SINCLAIR e TRS 80

SOFTWARES ESPECÍFICOS
Medicina, Odontologia, Laboratório de Análises Clínicas, Imobiliárias, Recrutamento de Pessoal e Assistência Técnica.

SOFTWARES NÃO ESPECÍFICOS
Rede Pert, Cadastro de Clientes/Mala Direta, Estoque/Inventário Permanente, Contabilidade, Editor de Textos, Contas a Receber, etc.

REVENDA
Equipamentos periféricos, calculadores, diskettes, fitas para impressoras, fitas K7 e formulários.

CURSOS
Basic e palestra para executivos.

PRESTAÇÃO DE SERVIÇOS
em bureau próprio e assessoria em HARDWARE e SOFTWARE para microcomputadores e sistemas IBM/370 e linha 43xx.

Visite nosso Show Room

MICRO'S

ASSESSORIA EM PROCESSAMENTO DE DADOS
Av. Indico, 71 • Fone: 448-6234 (011)
SÃO BERNARDO, SP • CEP. 09700

VIDEO K-7 E INFORMÁTICA

O Brasil finalmente entrou na era da eletrônica. Um dos sonhos dos brasileiros, o Video K-7, agora produzido por várias indústrias no Brasil, encontrou seu lugar nas empresas. Não são só os particulares que, usando e abusando desse extraordinário meio de comunicação, estão em sintonia com o mundo. Nas empresas começamos a encontrar, em larga escala, esse novo veículo. Sua utilização é muito ampla, reproduzir cartuchos com as últimas novidades de qualquer parte do mundo, sobretudo de algumas agências que se especializaram nesse ramo; gravar programas de grande interesse para sua posterior reprodução; para serem mantidos em arquivo, e agora, com a ajuda de uma Empresa totalmente brasileira, a SKILL INFORMÁTICA, pode ser utilizado o Video K-7 para o treinamento de pessoal. Esta é uma nova utilização, totalmente nova em nosso país, embora seja amplamente utilizada em outros lugares, onde o Video K-7 já está em uso a mais tempo.

A SKILL INFORMÁTICA denominou esse processo de STD ou seja, Sistema de Treinamento à Distância, pois os produtores do Sistema não se limitam, pura e simplesmente em vender o K-7, mas mantém um quadro de elementos capazes de dar toda a assessoria necessária às Empresas que optarem por seu sistema de treinamento STD.

O primeiro trabalho da SKILL é o STD DBaseII (um produto Ashton Tate representado no Brasil pela Datalógica), onde foi conseguido, com uma simplicidade espantosa, explicar-se toda a utilização desse fenomenal software que é o DBaseII.

Todas as Empresas que estão envolvidas com os sistemas de Processamento de Dados sabem muito bem dos problemas existentes para o adestramento da mão-de-obra especializada além do seu altíssimo custo.

O STD DBaseII produzido pela SKILL INFORMÁTICA, além de um custo muito inferior às 80 ORTNs (básico + avançado) normalmente cobradas para as aulas convencionais, pode ser diluído entre os vários técnicos e departamentos que dele farão uso, como também o STD DBaseII proporciona uma reciclagem periódica, uma facilidade de acesso e um cadastramento junto à SKILL, fazendo jus ao uso de sua Hot-Line para assessoria.

Este programa está gravado em duas fitas de formato VHS com duas horas de duração cada, envolvendo todas as informações sobre os comandos do software DBaseII, um dos programas mais vendidos nos Estados Unidos, pois segundo estatísticas recebidas, desde o lançamento até 1981 havia vendido 300.000 cópias.

Segundo informações prestadas pelos diretores da SKILL, Paulo Fernando Correa e Jesus Navarro, quando da comercialização do primeiro produto da Empresa, sentiram que o mercado está ávido para esse tipo de serviço e portanto já estão produzindo outras fitas como: "Introdução a Microcomputador", "Basic para CPM", "Basic para DOS" e "DBaseIII" entre outros.

Ronald Boock Stilck, Gerente de Vendas da SKILL, salientou que outro segmento que está em franca evolução são os grupos fechados que através de aulas convencionais atingem um desenvolvimento muito grande pois esses grupos não são restritos em termos de número mas sim em termos de interesse em função de ser dirigido às necessidades da empresa, tendo como didática o uso das aplicações reais para melhor assimilação por parte dos usuários, tanto do ponto de vista teórico, quanto prático.

Uma outra inovação, também inédita no mercado, é o treinamento convencional para o desenvolvimento de monitores em DBase com a finalidade de um assessoramento direto dentro das próprias empresas que utilizam o DBase envolvendo grande número de pessoas.

Ainda seguindo a mesma linha de pensamento, os Diretores da SKILL formaram um grupo de técnicos altamente capacitados para o desenvolvimento de aplicativos dentro dos softwares mais utilizados. Das características principais desses desenvolvimentos temos: são executados com o acompanhamento direto do cliente; carga horária real, de acordo com os aplicativos a serem desenvolvidos. Não se quer vender uma "Caixa Preta" que obrigue o cliente a recorrer continuamente à SKILL, quer-se criar a independência do mesmo dentro do manuseio do seu aplicativo e condições para que o cliente desenvolva outros aplicativos que necessite.

Sendo os produtos da SKILL de inteira confiabilidade, seus clientes tem seus serviços garantidos e só são debitados após sua total execução e satisfação e como diz Paulo Fernando Correa, "confiança não se pede, adquire-se".

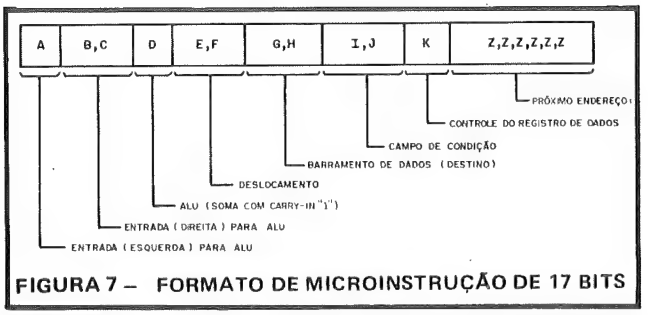


FIGURA 7 – FORMATO DE MICROINSTRUÇÃO DE 17 BITS

O problema geral de fornecer capacidade de ramificação de microprogramas é tratado em mais detalhes posteriormente, enquanto as várias alternativas para formatação e condificação de microinstruções são dadas na próxima seção.

O formato das microinstruções de 17 bits expandidas são dadas na figura 7. A função dessas microinstruções adicionais são como segue:

- **Condição do campo I, J**

Este campo especifica as condições que determinam os dois bits de endereço mais significantes W, Y (o endereço final tem a forma zzzzzzwy) para o próximo endereço de microinstrução.

00 : implementa bits W, Y para zero (00)
01 : implementa W = 0 e Y para o estado do registro de estado (OX)
10 : implementa W = 1, Y = 0 (10)
11 : implementa bits W, Y para um (11)

- **Controle de registro de estado K**

Este bit se encarrega de implementar o registro de estado para lembrar a condição de saída da ALU:

K = 0, nenhuma ação

K = 1, implementa o registro de estado para 1 se a lógica de detecção de zero detecta todos os zeros no barramento de dados; do contrário, ajusta-o para zero.

- **Próximo endereço Z (6 bits)**

Estes 6 bits alimentam diretamente a porção (zzzzzz) do endereço final para a próxima microinstrução a ser executada.

Deste modo, a primeira das três microinstruções deve ser modificada como se segue: o bit do registro de estado deve ser K = 1 para que o registro de estado fique "setado", se o resultado de A – B for zero. Os bits de condição deverão ser 01, para que o bit mais significante Y do próximo endereço leve no valor correspondente o resultado da primeira microoperação. A figura 8 dá o diagrama em blocos detalhado da figura 5, mostrando a forma binária das microinstruções necessárias à execução das respectivas microoperações e indicações de endereço de microinstrução.

Neste exemplo definimos o fluxo de dados através da seção de processamento, que executa a macroinstrução "COMPARE" e tem determinado o que a microinstrução pode fazer diretamente, sem otimizar a execução ou reduzir as necessidades de temporização.

FORMATAÇÃO E CODIFICAÇÃO DE MICROINSTRUÇÕES

A memória microprogramada de uma máquina bit sliced é implementada usando-se ROM's, PROM's, EPROM's ou PLA's. No entanto, o projetista pode escolher entre uma grande variedade de formatos e técnicas de codificação na estruturação de suas microinstruções.

Na sua forma mais simples, uma microinstrução sempre possui duas partes funcionais; uma constituída de bits correspondentes aos padrões de sinais de controle, que definem e controlam todas as microoperações elementares a serem executadas; e outra constituída de bits que identificam e controlam o endereço da próxima microinstrução a ser executada, por exemplo, identificam a seleção da fonte do próximo endereço de microinstrução ou algumas vezes fornecem o endereço real da próxima microinstrução.

Em geral, há uma grande variedade de formatos de mi-

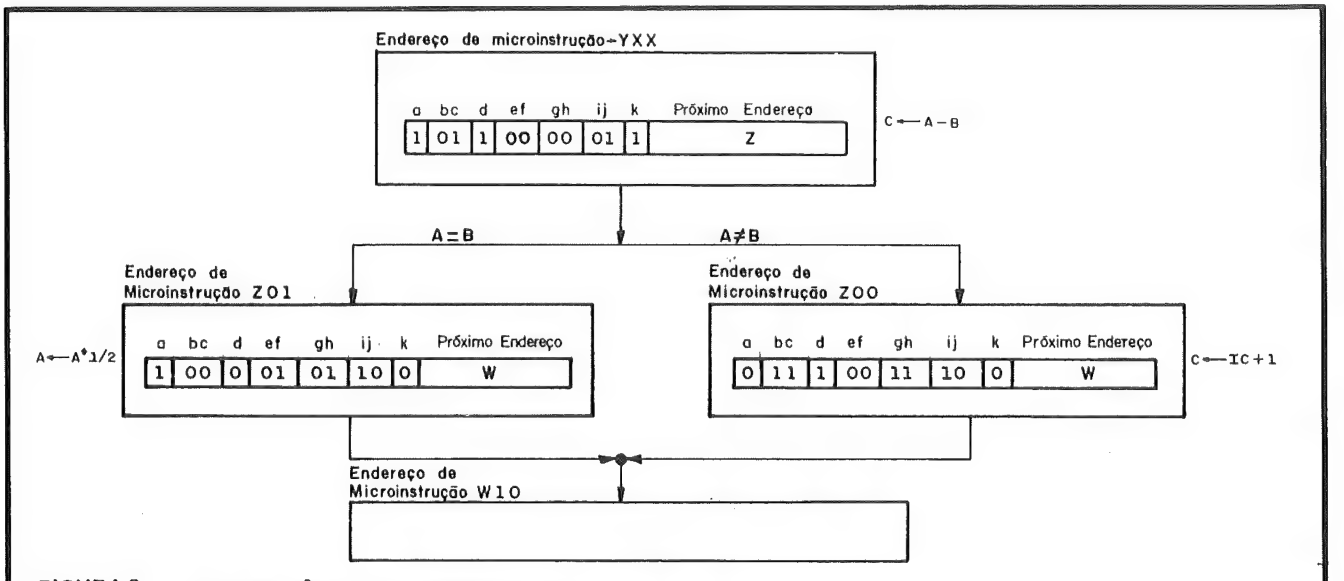


FIGURA 8 – TRANSFERÊNCIA DE ENDEREÇO DE MICROINSTRUÇÃO E ESTRUTURA DE CADA MICROINSTRUÇÃO PARA EXECUÇÃO DA INSTRUÇÃO "COMPARE"

croinstruções, extensões e esquemas de temporização. Quando utilizamos máquinas bit sliced, no entanto, o uso de RALU's pré-definidas (slices da seção de processamento) limita as opções. A formatação sempre se refere à definição de vários campos da microinstrução.

O formato da microinstrução pode ser fixo ou variável. No formato fixo, cada bit da microinstrução é sempre interpretado do mesmo modo, enquanto que no formato variável o significado dos bits ou campos muda de acordo com certas condições do sistema. O formato variável leva a microinstruções mais curtas, mas também requer lógica adicional e um tempo mais longo no controle do formato.

Se cada bit na microinstrução é usado somente para controlar uma linha de controle separada ou apontar no sistema, então, a microinstrução é chamada "não-codificada", "desempacotada", ou ainda, "forma explodida", vide figura 9. Este formato leva a palavras de microinstrução mais compridas, velocidades de operações maiores e grande coincidência. É claro que uma microinstrução mais ampla sempre aumenta a memória do microprograma.

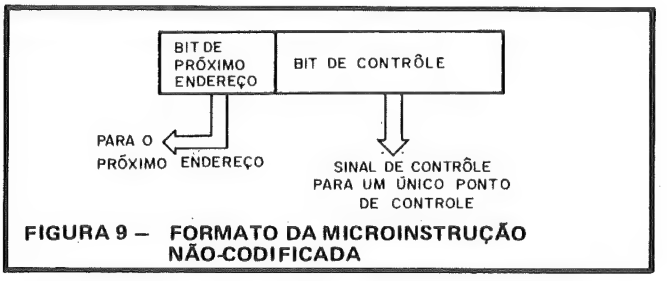


FIGURA 9 – FORMATO DA MICROINSTRUÇÃO NÃO-CODIFICADA

MICROPROGRAMAÇÃO VERTICAL E HORIZONTAL

Na figura 10, um esquema alternado identifica grupos de linhas de controle mutuamente exclusivas que são controladas por um único campo de controle codificado na microinstrução. Se, por exemplo, 2^n linhas existem, um campo de **n** bits é necessário na microinstrução. Esses **n** bits, por meio de um decodificador, realimentam 2^n sinais de controle quando a microinstrução é executada. Tal microinstrução é chamada **codificada** ou de **forma empacotada.** Dois grupos de linhas de controle, que são mutuamente exclusivas, devem ser controladas por dois campos separados de controle codificados da microinstrução. Os campos de controle separados são também fornecidos para cada unidade funcional independente, tais como ALU, deslocadores etc. Isso permite o uso simultâneo de recursos do sistema, contanto que eles não façam uso conflitante de um conjunto de dados. Esse esquema leva a microinstruções mais curtas e memórias de microprograma menores.

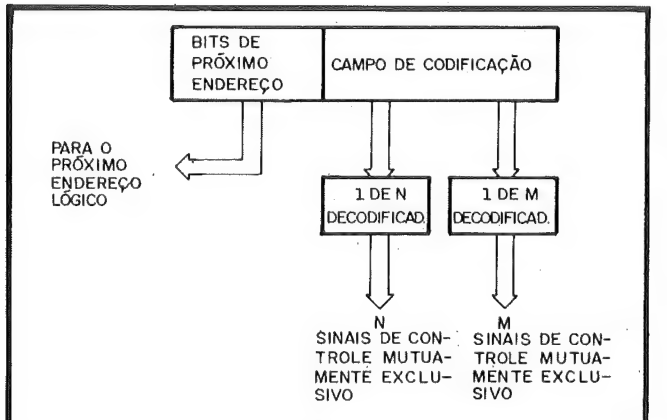


FIGURA 10– FORMATO DE MICROINSTRUÇÃO CODIFICADA COM LINHAS DE CONTROLE MUTUAMENTE EXCLUSIVAS.

Dois outros termos comumente usados são microprogramação "vertical" e "horizontal". Elas são bastante similares aos termos codificados e não-codificados, respectivamente, embora nenhuma definição universal desses termos exista. Geralmente, microprogramação horizontal refere-se ao exercício do controle direto, ou quase direto, sobre o hardware. Esta habilidade conduz a microinstruções mais amplas com muitos campos de controle separados. Ao mesmo tempo, porém, há um mínimo de codificação, aumento de pontos de convergência e um número menor de microinstruções no microprograma; deste modo, fornecendo uma técnica poderosa para implementar algoritmos de administração de recursos. Essa precisa associação de microcodificação horizontal com hardware foi também descrita como **microprogramação estrutural.**

Na microprogramação vertical existe um alto estágio de codificação que de certo modo remove as microinstruções do hardware. O repertório de microinstruções torna-se menor, mas a sua habilidade em operar simultaneamente decresce; a complexidade da microinstrução aumenta e a velocidade efetiva do computador decresce, requerendo vários ciclos internos de clock para mecanizar a execução das microinstruções. Este formato também é conhecido pelo nome de **microprogramação funcional.**

Alguns computadores têm outro modo de operação na qual uma microinstrução codificada, através de um número limitado de campos de controle, endereça ainda outro nível menor de armazenamento chamado **nanostore,** figura 11, contendo nanoinstruções. A **microstore** (memória do microprograma) contém um grande número (1K) de palavras pequenas (8 bits), enquanto a **nanostore** é larga, mas contém menos palavras (256). Embora a memória do microprograma total de 4K de palavras de 32 bits possa ser implementada com apenas 4K de palavras de 8 bits mais 256 palavras de 32 bits, um meio fácil de implementar dois níveis de **microstore** é através do uso de FPLA's (Field Programmable Logi Arrays). As nanoinstruções correspondem às sub-rotinas no microprograma. Dois exemplos significativos da implementação da nanoprogramação

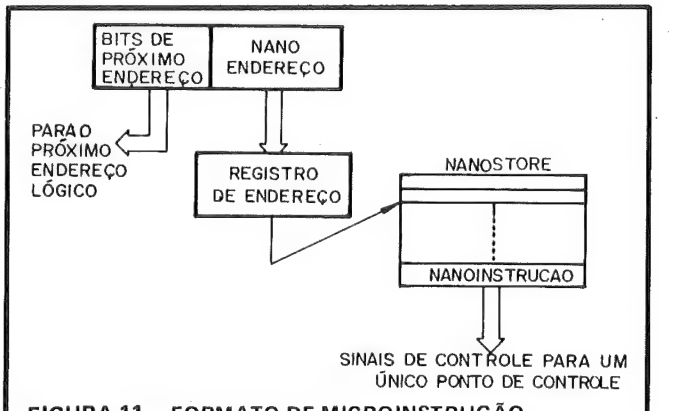


FIGURA 11– FORMATO DE MICROINSTRUÇÃO NANOPROGRAMADA

são o Nanodata QM-1 e o CII MITRA. A figura 12 apresenta um projeto alternativo de dois níveis de memória usado com microinstruções horizontais.

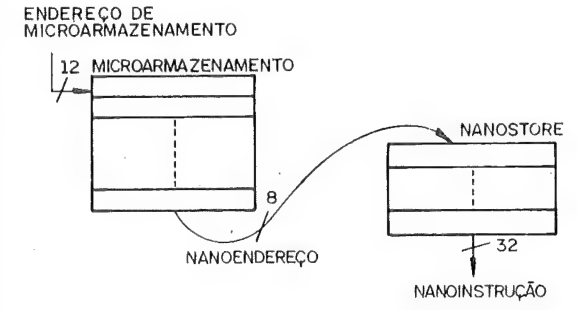


FIGURA 12 – FORMATO DE MICROINSTRUÇÃO NANOPROGRAMADA COM DOIS NÍVEIS DE CONTROLE DE MEMÓRIA.

Dois outros termos comuns relacionados com a temporização da microinstrução são a microprogramação **monofásica** e **polifásica**. A microprogramação monofásica é a mais simples das duas, pois cada microinstrução requer apenas um único ciclo de clock, figura 12. Ela requer mais microinstruções na memória microprogramada, mas sua extensão é menor. Por outro lado, a microprogramação polifásica é mais complexa porque a microinstrução requer mais que um único ciclo de clock, isto é, sinais de controle de microinstrução são emitidos em seqüência sobre vários ciclos de clock, figura 14. Os sistemas polifásicos permitem sobrepor as funções do microprocessador, permitindo um aumento nas velocidades efetivas de operação. A atribuição de atividades a cada **slot** de tempo aumenta a capacidade nos microprocessadores **bit sliced**.

Tanto na monofase como na polifase a estrutura de temporização pode também variar quando alguns campos de controle na microinstrução têm informações de temporização explícitas ou implícitas associadas a eles.

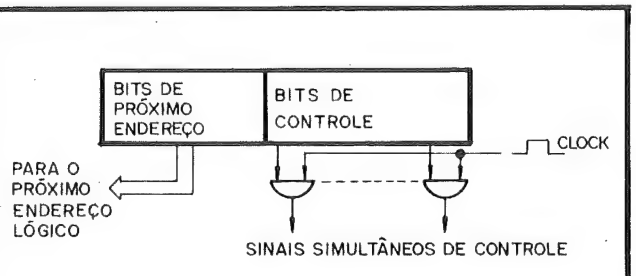


FIGURA 13 – FORMATO DE MICROINSTRUÇÃO MONOFÁSICA

Na escolha do formato da microinstrução o projetista tem a tarefa de avaliar o tamanho da memória de microprograma, esquemas de clocking, velocidades, flexibilidade de desvios e a complexidade do cirucito de saída necessário para manipular os bits de controle das microinstruções ou decodificar os campos das microinstruções. Em geral, as características das microinstruções estão fortemente influenciadas pelo projeto de fluxo de dados da seção de processamento e pelo conjunto de instruções de linguagem de máquina que a seção de processamento apóia. Em projetos anteriores, a organização de uma seção de processamento eficiente envolvia transações complexas. Atualmente, com o uso das RALU's comercialmente disponíveis (slices da seção de processamento), decresce a forte influência que o projeto de fluxo de dados da seção de processamento tinha sobre as características das microinstruções.

Em geral, podemos dizer que em máquinas com microinstruções verticais a implementação é monofásica com execução serial. Com o aumento do tamanho da microinstrução uma combinação serial-paralelo se torna atraente. Em máquinas com microinstruções horizontais a implementação é polifásica com execução justaposta. Johnson e Merwin conduziram uma avaliação comparativa, a qual mostrou que para otimização da velocidade de execução o formato da microinstrução horizontal fornecia vantagens definitivas para simples programas seqüenciais e interativos. No entanto, poucas máquinas de propósito geral utilizam microinstruções que possam ser classificadas como puramente horizontais.

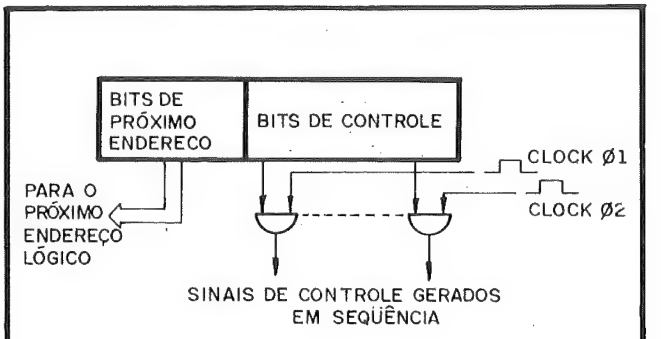


FIGURA 14 – FORMATO DE MICROINSTRUÇÃO POLIFÁSICA

ASPECTOS DE MICROPROGRAMAÇÃO

Temos nos referido neste artigo aos microprocessadores bit sliced como sendo máquinas microprogramáveis pelo usuário "dinamicamente", onde "dinamicamente" indica máquinas contendo memória microprogramada, que podem ser carregadas sob o programa de controle; e "micropro-

Dois grupos de linhas de controle, que são mutuamente exclusivas, devem ser controladas por dois campos separados de controle codificados da microinstrução.

gramáveis pelo usuário", que implica em oferecer ao usuário as ferramentas necessárias e o apoio para lidar com ambos os aspectos de hardware e software de microprogramação. O uso da memória microprogramada e a disponibilidade de sofware de apoio caracterizam a facilidade com que a máquina pode ser microprogramada pelo usuário.

Lidando com os aspectos de hardware da microprogramação o usuário tem a necessidade de conhecer a arquitetura da máquina. Depois disso, ele precisa avaliar as necessidades conflitantes para chegar ao melhor formato da microinstrução para esta aplicação. Tais formatos são, até certo estágio, ditados pelo tipo específico de slice do processador escolhido.

Do ponto de vista de software, a microprogramação pode ser vista como sendo conceituadamente similar à programação, com a principal diferença quanto ao nível de controle exercido. O saftware de apoio disponível ao microprogramador varia; para microprocessadores é mais pobre que para minicomputadores. Mesmo para minicomputadores. , o desenvolvimento de linguagens de microprogramação retarda-se ao desenvolvimento das linguagens de programação.

O software de apoio para microprogramação inclui "tradutores" para transformar as representações de alto nível em microprogramas (algumas vezes em linguagem intermediária que é interpretada em microcódigo), pronto para carregar na memória do microprograma "simuladores" para ajudar no **debugging** detalhado dos microprogramas e **programas de debugging on-line** para ajudar na monitoração de microprogramas em execução.

Em geral, as linguagens de microprogramação podem ser classificadas em linguagens de baixo nível e linguagens de alto nível, como mostra a seguinte tabela:

BAIXO NÍVEL

Microlinguagens
Linguagens de microprogramação assembler
Linguagens de microprogramação de fluxograma
Linguagens de microprogramação de transferência de registro

ALTO NÍVEL

Linguagens de microprogramação de transferência de macro registro
Linguagens de microprogramação dependentes da máquina de procedimento orientado
Linguagens de microprogramação independentes da máquina de procedimento orientado.

IMPLEMENTAÇÕES USANDO COMPONENTES MSI

Os seqüenciadores de microprogramas podem ser implementados por simples controle seqüencial sem capacidade de desvio. Em uma das formas mais simples de seqüenciadores de microprogramas a única coisa requerida para avançar sucessivamente através das microinstruções é um contador de endereços de microprogramas, figura 15A. O endereço da próxima microinstrução é selecionado simplesmente incrementando uma unidade ao contador de endereço de microprograma em cada ciclo do gerador de sinais de sincronismo. A microinstrução, por ela mesma, não contém um campo de "próximo endereço". Esta técnica de contagem permite apenas microcontrole seqüencial; ele não é muito flexível e não proporciona meios de se alterar, condicional ou incondicionalmente, o fluxo de controle estabelecido.

Os seqüenciadores de microprogramas podem ser usados para seqüenciamento com capacidade de desvio incondicional. No esquema de contador de endereço do microprograma o contador pode ser deslocado para o endereço zero apertando a entrada **clear** ou, quando acionado pelo **clock**, pode carregar um novo número que é apresentado nas linhas de entrada de dados, proporcionando assim capacidade de desvio para um novo endereço.

A figura 15B mostra uma outra configuração similar usando um registro de endereço de microprograma, que é

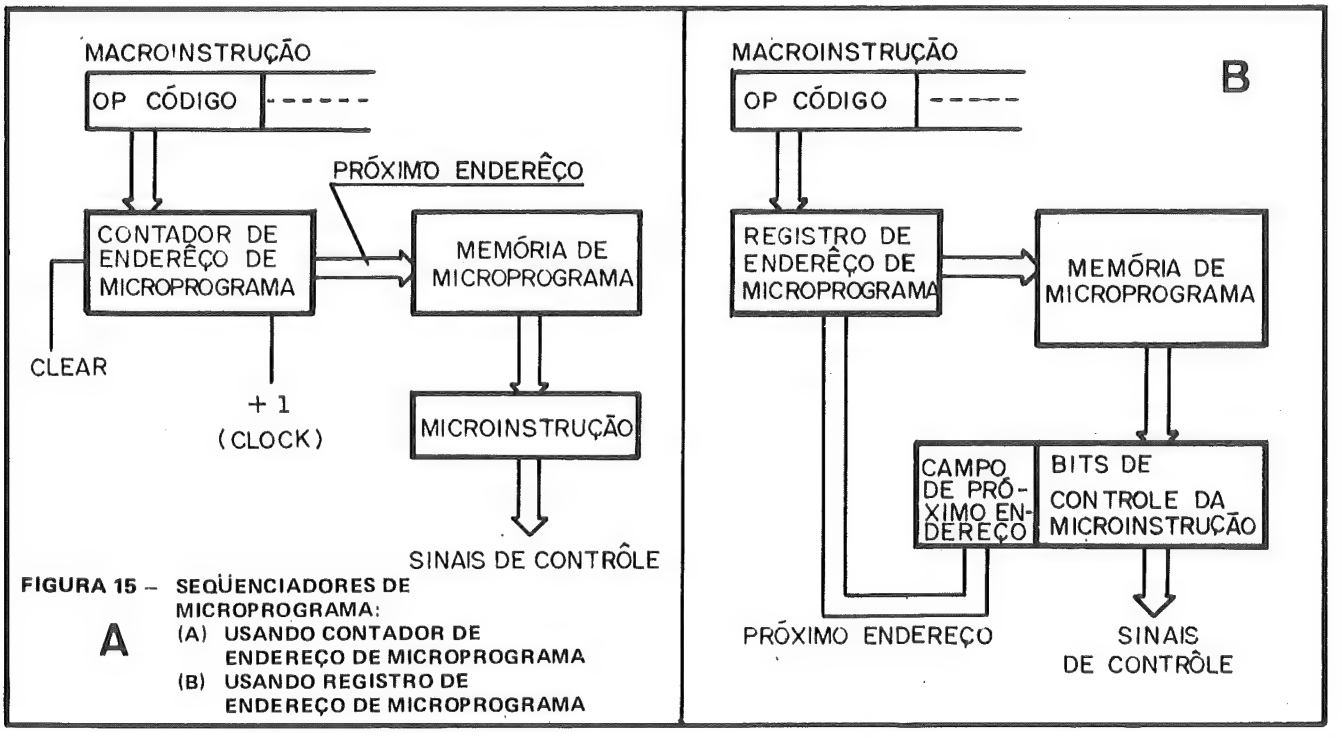


FIGURA 15 – SEQÜENCIADORES DE MICROPROGRAMA:
(A) USANDO CONTADOR DE ENDEREÇO DE MICROPROGRAMA
(B) USANDO REGISTRO DE ENDEREÇO DE MICROPROGRAMA

carregado com o próximo endereço de um campo específico da microinstrução em execução.

As microinstruções, agora, tornam-se maiores, mas oferecem capacidade adicional de fazerem "saltos" (jumps) incondicionais no microprograma. Novamente, isso não proporciona meios de se alterar condicionalmente o fluxo de controle estabelecido. Esta configuração é útil quando o programa a ser implementado tem várias condições de "saltos" incondicionais e quando o contador deve ser minimizado.

Já que os dois esquemas anteriores não apresentam a capacidade para execução de desvio condicional (necessário para a execução repetida de **loops**), seqüenciadores de microprogramas podem ser usados para seqüenciamento com capacidade limitada de desvio condicional. A técnica de modificação do contador, mostrada na figura 16, pode implementar uma simples facilidade de desvio condicional, onde as microinstruções contém um **bit de controle** especial. Se este bit de controle é zero, o endereço do contador de microprograma é incrementado de uma unidade. Se, contudo, o bit de controle for um, então o endereço do contador de programa estará liberado (LD) e o próximo endereço (que poderá ser um desvio) contido na microinstrução é carregado paralelamente no endereço do contador de microprograma. Como extensão do que foi mencionado anteriormente, pode-se considerar que se o tamanho do campo do bit de controle for **n** bits poderemos obter desvio condicional de 2^n condições diferentes. Este campo expandido de bit de controle, quando decodificado, determina o instante em que o próximo endereço poderá estar sendo carregado paralelamente no contador de endereço do microprograma.

Até agora, apresentamos seqüenciadores de microprograma com capacidade limitada de desvio condicional. Eles poderão ter maior capacidade de criar decisões se puderem examinar, antes do desvio, a condição ou estado dos bits que a seção de controle originou.

Os diagramas de bloco, figuras 17, 18 e 19, mostram

As microinstruções, agora, tornaram-se maiores, mas oferecem capacidade adicional de fazerem "saltos" incondicionais no microprograma.

três maneiras de implementar capacidade de desvio baseado em diferentes condições externas. Na figura 17, a lógica de decisão analisa o estado do **flag** (proveniente da seção de processamento) e os bits do campo de controle da microinstrução, verificando se a condição dada encontra-se em posição para:

- continuar seqüencialmente;

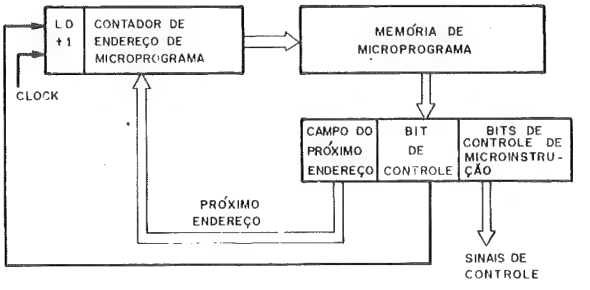


FIGURA 16 – MODIFICAÇÃO NO SEQUENCIADOR DE MICROPROGRAMA USANDO CONTADOR PARA PERMITIR CAPACIDADE DE SIMPLES DESVIO CONDICIONAL.

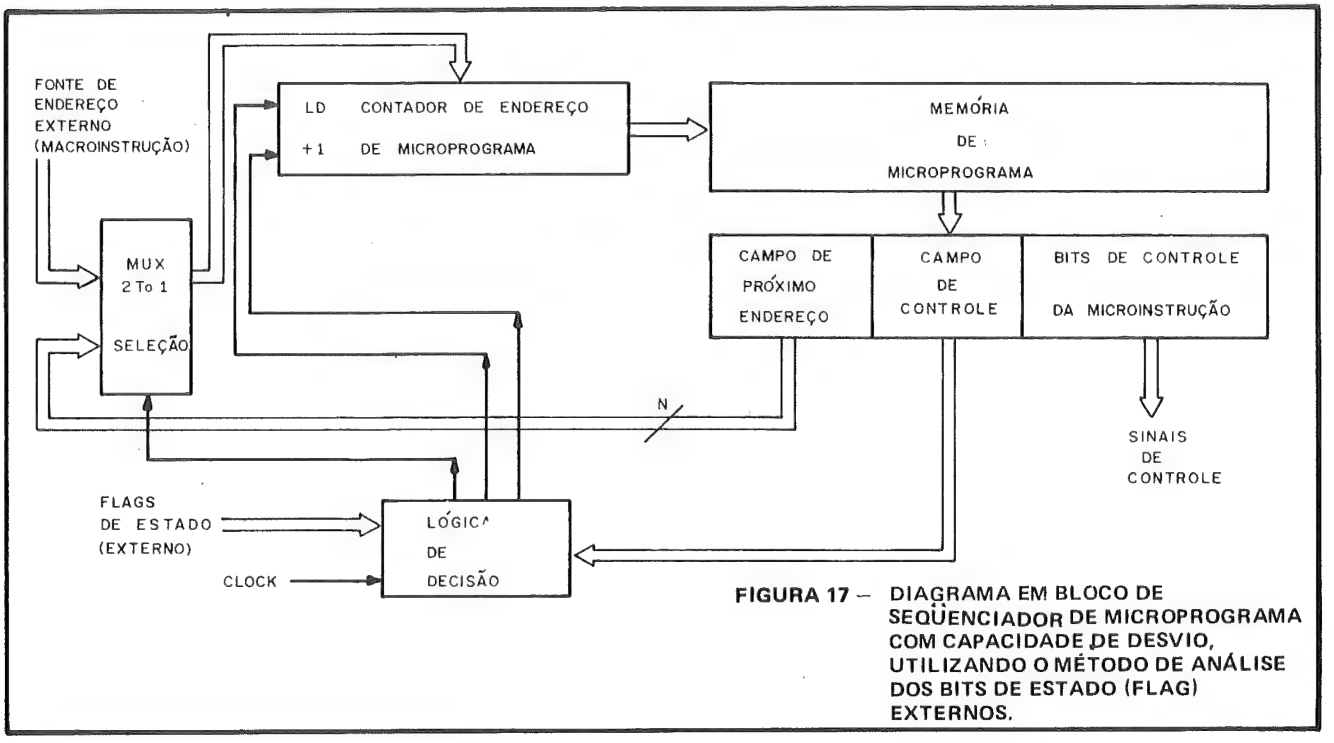


FIGURA 17 – DIAGRAMA EM BLOCO DE SEQÜENCIADOR DE MICROPROGRAMA COM CAPACIDADE DE DESVIO, UTILIZANDO O MÉTODO DE ANÁLISE DOS BITS DE ESTADO (FLAG) EXTERNOS.

- desviar para a microinstrução apontada pelo próximo campo de endereço da microinstrução corrente;
- substituir o próximo endereço por aquele fornecido por uma fonte externa (código de operação da macroinstrução).

Na figura 18, a configuração fornece a capacidade para desvio em cada uma das duas diferentes condições externas, C1 e C2. Os dois bits do campo de controle da microinstrução selecionam uma das quatro condições de entrada do **multiplex** para controlar a entrada **LD** do contador de endereço do microprograma. Duas dessas entradas do **MUX**, I1 e I2, representam duas diferentes condições externas que, quando verdadeiras, causam o desvio condicional carregando paralelamente o próximo en-

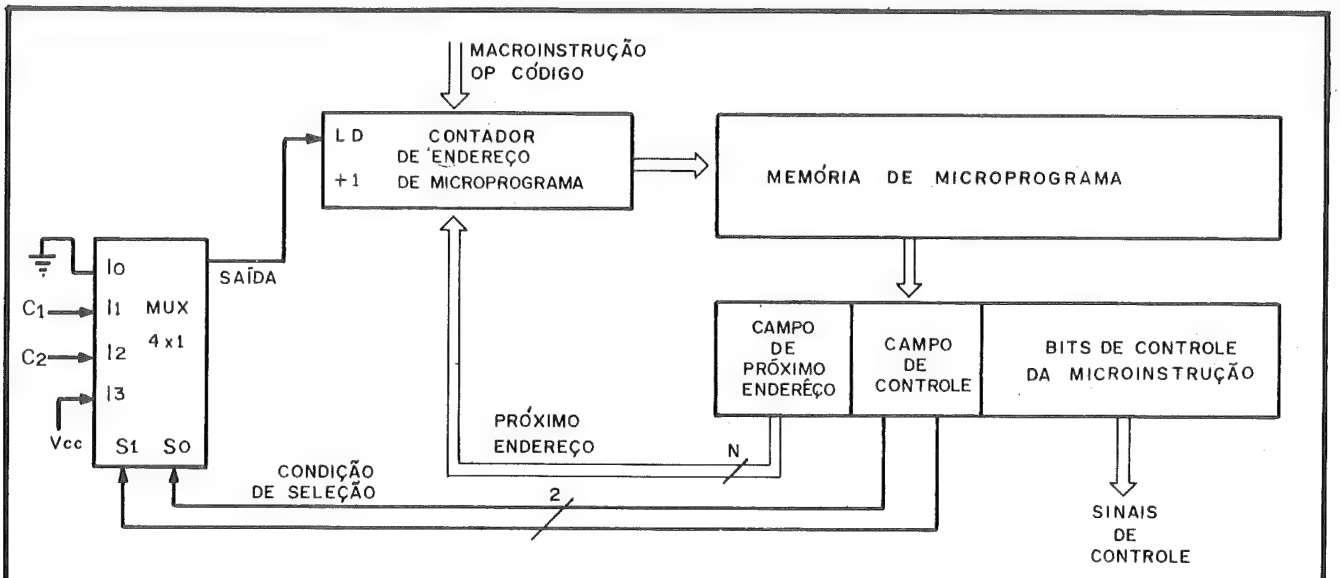


FIGURA 18 – DIAGRAMA EM BLOCO DE SEQÜENCIADOR DE MICROPROGRAMA COM CAPACIDADE DE DESVIO, UTILIZANDO O MÉTODO DE ANÁLISE DE DUAS CONDIÇÕES EXTERNAS.

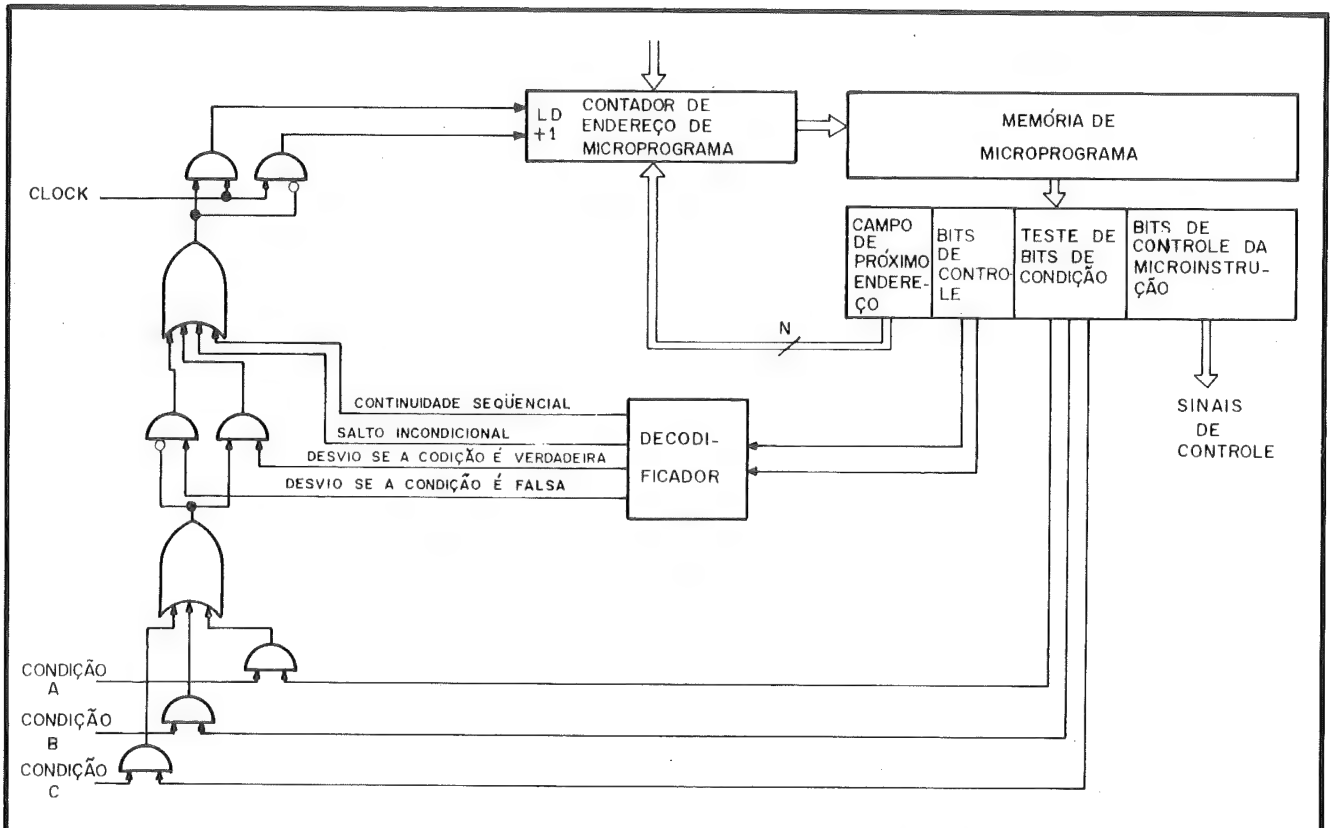


FIGURA 19 – DIAGRAMA EM BLOCO DE SEQÜENCIADOR DE MICROPROGRAMA COM CAPACIDADE DE DESVIO UTILIZANDO O MÉTODO DE ANÁLISE DE TRÊS CONDIÇÕES EXTERNAS.

dereço na microinstrução corrente no contador de endereço do microprograma. A entrada **10**, que corresponde ao campo de controle da microinstrução **00**, força o **LD** inativo e, desse modo, o microprograma continua seqüencialmente para a próxima microinstrução. A quarta entrada I3, que corresponde ao campo de controle **11**, efetivamente causa um "salto" incondicional. Finalmente, percebe-se que a figura 19 é um tanto similar à figura 18, exceto as três condições externas A, B e C e o seqüenciador de microprograma dessa configuração, que é implementado usando componentes SSI.

A figura 20 mostra exemplos de seqüenciamento de microprogramas envolvendo técnicas de desvio condicional dentro da mesma página de microprograma. Na figura 20 A a microinstrução INC executada na posição **X** é usada para seqüenciar seriamente o registro de endereço do microprograma de uma unidade. Quando a condição **TEST** examinada por **BRT** é verdadeira o desvio é executado. O desvio do endereço é fornecido via campo do próximo endereço da microinstrução na posição **X + 1**. Nesse exemplo, o desvio do endereço é **Y**. No equivalente **ALGOLTYPE** a notação **MX** indica a microinstrução que está para ser executada em paralelos como teste especificado. Desse modo, na figura 20A a microinstrução **M0** tem que estar apta para alcançar **M1** (se A é verdade) ou **M2** (se A é falso) no próximo ciclo de clock. Este pode ser terminado posicionando o endereço de **M1** (Y) no campo do próximo endereço de M0 e posionando a microinstrução **M2** imediatamente depois de **M0** (na posição X + 2). **M3** é o sucessor para a seleção da seqüência (M0 – M2). A analogia é mantida para o exemplo da figura 20B, onde a microinstrução **M1** está em **Y** e com os campos de próximo endereço apontados para o endereço **Z** da microinstrução **M2**. Similarmente, **M4** está em X + 2, enquanto **M6** está em X + 3. Esse exemplo representa implementações de desvio condicional, usando a técnica chamada de **Desvio Binário**. Essa técnica não é aconselhável para condições complexas visto que requer o uso de muitos desvios condicionais sucessivos.

IMPLEMENTAÇÃO DE SEQUENCIADORES COM FPLA (FIELD-PROGRAMABLE LOGIC ARRAYS)

Os circuitos apresentados nas partes anteriores deste artigo são particularmente simples, porém, para situações mais práticas eles podem tornar a microinstrução bem extensa. Uma maneira de contornar essa desvantagem é alimentar os **N bits** do campo de próximo endereço da microinstrução com uma **PROM** ou **FPLA**, contendo M bits (M > N), que é aplicada ao contador de endereço de microprograma para carregamento paralelo. Isto corresponde a dois níveis de implementação de memória de

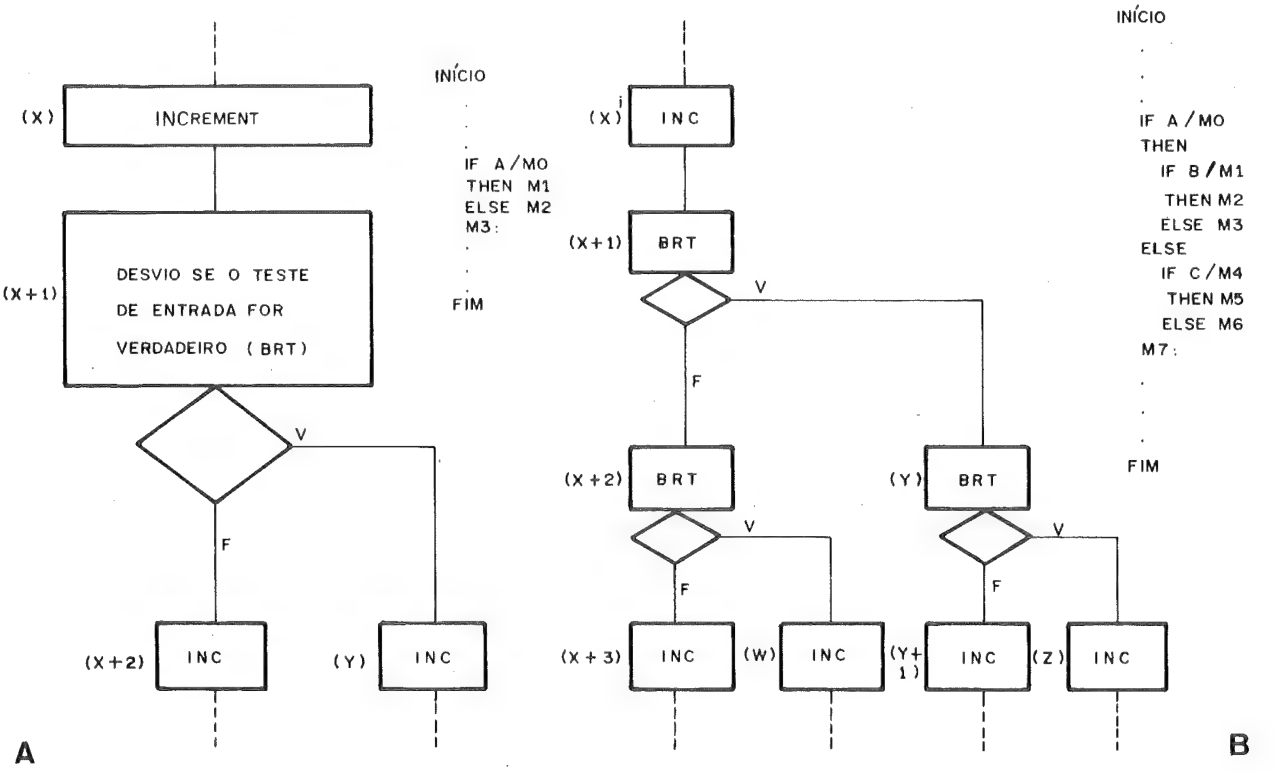


FIGURA 20 – FLUXOGRAMA E SUAS RESPECTIVAS ESTRUTURAS EM ALGOL APRESENTANDO DOIS DESVIOS CONDICIONAIS:

A) TESTE SIMPLES E DESVIO CONDICIONAL

(B) TESTE MAIS COMPLEXO E DESVIOS CONDICIONAIS

microprograma: um **microstore** e um **nanostore**. Contudo, essa técnica diminui a velocidade de processamento, visto que esta implementação requer um ciclo de acesso extra e torna a microprogramação ainda mais difícil, visto que envolve o manejo de dois espaços de endereço.

Uma **FPLA** pode ser vista como uma implementação LSI da clássica estrutura **Soma-de-Produtos**. A figura 21 mostra um diagrama simplificado de uma FPLA en-

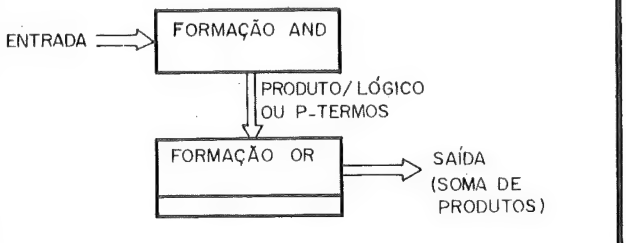


FIGURA 21 – DIAGRAMA DE UMA FPLA

quanto a figura 22 apresenta mais detalhes da estrutura interna. O primeiro nível **AND ARRAY** (formação **AND**) executa a lógica **AND** para combinações selecionadas de sinais da entrada.

Estas portas **AND** executam o **produto lógico ou P-termos**: $P_1, P_2, \ldots, P_m$, onde "m" é tipicamente 48 ou 96. O segundo nível **OR ARRAY (formação OR)** é composto de portas **OR** que executam a **soma lógica** de alguns ou todos os **P-termos** produzidos pelo primeiro nível, fornecendo uma saída final ou soma de termos: $F_1, F_2, \ldots F_k$, onde K é tipicamente 8.

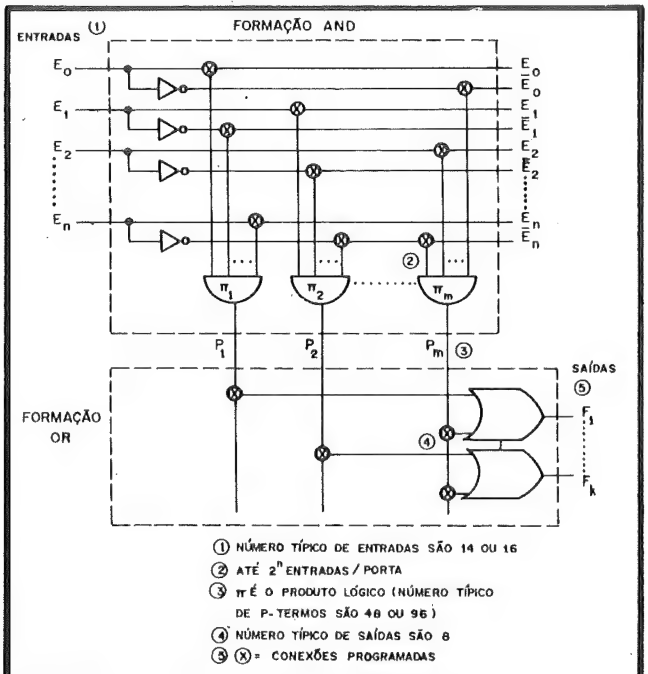


FIGURA 22 – ESTRUTURA INTERNA DE UMA FPLA

Comparadas às **PROM'S** (Programmable Read Only Memory) as **FPLA's** são mais eficientes porque permitem um grande número de entradas. Um chip **PROM** pode ser considerado como composto de um decodificador, que transfere os endereços de entrada para uma localização da memória, e uma parte de memória para armazenamento da informação. Enquanto a parte de memória é programável, a parte do decodificador é inalterável.

As FPLA's diferem das **PROM's** na parte relativa ao decodificador que também pode ser programável. Na implementação de circuitos combinacionais com **FPLA's** podemos ir diretamente da função lógica para a programação das formações **AND** e **OR**.

Esta vantagem torna-se mais clara nos casos em que se tem um grande número de entradas, isto é, muitas variáveis. Nesses casos, escrever uma tabela-verdade seria um trabalho árduo.

A **FPLA** é sugerida quando houver um grande número de entradas, porém, somente se um pequeno subconjunto dessas combinações possíveis de entrada forem necessárias. Assim sendo, pode-se usá-la na seção do **microprocessador bit sliced** para a decodificação da macroinstrução; para decodificação do hardware e do programa de **status**; como uma memória de microprograma quando o conjunto de instruções utiliza um pequeno número de palavras e longos códigos de operação; e para decodificação de microinstrução.

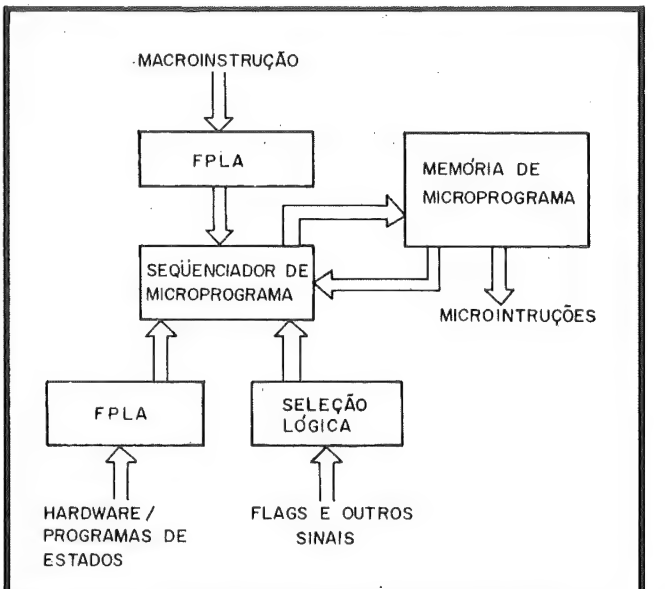


FIGURA 23- FPLA's USADAS PARA DECODIFICAÇÃO DE MACROINSTRUÇÃO, HARDWARE E PROGRAMAS DE ESTADOS

A figura 23 apresenta uma configuração onde as **FPLA's** são usadas para decodificação da macroinstrução e para decodificação do hardware e do programa de **status**.

O micro NAJA foi desenvolvido utilizando os mais modernos padrões de arquitetura de Microcomputador, atingindo uma ampla faixa, desde os computadores pessoais até os utilizados em empresas de pequeno e médio porte. Uma de suas grandes vantagens é a sua versatilidade, ou seja, você poderá adquiri-lo na sua versão mais simples, podendo você mesmo expandi-lo à medida de suas necessidades, a um baixo custo

CARACTERÍSTICAS TÉCNICAS

- *48K bytes de memória RAM*
- *16K bytes de memória ROM*
- *Clock de 3,6 MHz ou 2,1 MHz comutado por Soft*
- *Saída para impressora paralela*
- *6 conectores para expansão no próprio gabinete*
- *Microprocessador Z-80A*
- *Vídeo de 16 linhas por 64 ou 32 colunas*
- *Interface de cassete para 1.500 ou 500 BPS*
- *Linguagem Basic na ROM do sistema*
- *Software compatível com TRS-80 mod. III*

ACESSÓRIOS

- *Monitor de vídeo de 12" verde profissional*
- *Interface para 4 unidades de disco de 51/4" de dupla dens.*
- *Unidade de disco face simples ou dupla*
- *Interface para 4 MHz de Clock*

AGORA TAMBEM COM:

- CP/M
- CLOCK DE 6 MHZ
- UNIDADE DE DISCO DE 8"
- VIDEO A CORES
- SINTETIZADOR DE VOZ

KEMITRON LTDA.

Av. Contorno, 6048 - Savassi - Fone: 225-0644
Telex - (031) 3074 - KEMI - BR Belo Horizonte - MG

A CLAPPY VAI MEXER NO SEU EGO

Se você sonhava com um computador EGO em sua empresa, agora a Clappy oferece as condições ideais, tanto em termos de custo e formas de pagamento, como, principalmente, nos diversos serviços que a Clappy pode prestar.

Assistência técnica, treinamento, implantação e instalação de sistemas, cursos de operação e programação, consultoria, além de grande variedade em software, periféricos, suprimentos, aplicativos comerciais e de apoio.

Tudo isso para que seu EGO seja utilizado em toda a sua potencialidade.

EGO-HARDWARE

O EGO, tem memória expansível até 1 milhão de bytes. Podem ser conectados até 4 disquetes de 5 1/4 ou 8" e 2 discos Winchester de 10 megabytes cada.

O brasileiro de 16 Bits mais compatível com o IBM PC.

EGO-SOFTWARE

O EGO opera com todas as alternativas de Sistemas Operacionais: ANALIX, CPM 86 e MS DOS e Linguagens disponíveis: BASIC, COBOL ANS 74, PASCAL, FORTRAN, APL, MUMPS e C, atendendo desde a pequena empresa até grandes organizações, adequando-se às características de cada uma.

EGO - O COMPUTADOR PROFISSIONAL

Os recursos do EGO atendem até os usuários mais exigentes. É um computador multiusuário que não tem as limitações dos demais micros e garante a performance dos mini. É compatível com o IBM-PC e tem processador de 16 Bits. Com certeza um EGO que faz muito bem ao seu.

EGO - TERMINAL IBM

O EGO pode emular terminais IBM da família 3270, inclusive o terminal gráfico colorido 3279, e suporta protocolos assíncronos BSC nível 3 e nível 1.

APLICATIVOS DE APOIO À DECISÃO

Planilha Financeira, Processamento de Texto, Mala Direta, Cadastro de Clientes, Controle Financeiro, etc...

APLICATIVOS COMERCIAIS

Folha de Pagamento e Contabilidade.

A Clappy desenvolve soluções específicas para qualquer problema que você tenha.

Venha à nossa loja ou solicite a visita de um representante.

SOFTEC ENG. SISTEMAS COM. LTDA.

Clappy

Centro: Av. Rio Branco, 12 - Loja e sobreloja - Tel.: (021) 253-3395

Copacabana: Rua Pompeu Loureiro 99 - Tels.: (021) 236-7175 e 257-4398

Aberta diariamente das 9 às 19 horas e aos sábados das 9 às 14 horas.
Estacionamento próprio.
Entregamos em todo Brasil pelo reembolso Varig.

Substituir a memória do microprograma da seção de controle por uma ou mais **FPLA**'s nada mais é do que implementar uma memória somente de leitura de índice de endereço, onde os desvios condicionais são mais "vindo de" do que "indo para".

SEQUENCIADORES COMERCIAIS DE MICROPROGRAMA LSI

Atualmente, os seqüenciadores comerciais de microprograma **LSI** possuem todas as características mencionadas anteriormente e mais algumas. Eles permitem um maior número de sinais de controle e sinais de bits de estado e possuem os requisitos necessários para implementação de funções de incrementação de endereço e complexo microdesvio condicional. Também possuem um contador de **loop** no próprio **chip** para **loops** de microprogramas repetidos e incluem "pilhas de ponteiros" (STACK POINTER) que garantem o retorno dos endereços das micro sub-rotinas. O diagrama em bloco de um seqüenciador lógico LSI comercial é aparesentado na figura 24 onde o próximo endereço lógico é mais complexo que os mencionados anteriormente.

O principal objetivo de um seqüenciador de microprograma é apresentar um endereço para a memória de microprograma, de maneira que a microinstrução possa ser buscada e executada.

Parte do próximo endereço lógico do seqüenciador determina a fonte do endereço específico a ser carregada no registro contador de endereço do microprograma. A escolha da fonte de endereço é baseada nos bits de informação do próximo endereço que o seqüenciador recebe. Geralmente esses bits podem especificar uma das seguintes informações: incremento, salto condicional da próxima instrução, desvio condicional para uma sub-rotina, **Push/Pop** na "**pilha**" e assim por diante. Muitos desses seqüenciadores LSI incluem no chip um registro acumulador, com quatro níveis de profundidade, operando na forma LIFO (Last In First Out – último a entrar, primeiro a sair) Esse acumulador é usado como armazenador temporário de endereços durante um **loop** de microprograma ou durante um desvio da micro sub-rotina, operando sob um comando tipo **Push** e **Pop**.

Os seqüenciadores de microprograma LSI, disponíveis no mercado, podem ser divididos em duas categorias: aqueles com um número fixo de posições de memória dentro de sua capacidade de endereço e aqueles que são dispositivos **bit slice** expandíveis em ligações em cascatas. Como membros da primeira categoria temos o INTEL 3001, figura 25, as MONOLITHIC MEMORIES 67110, o SIGNETICS 8 X 02, figura 27, e o FAIRCHILD 9408.

TENHA O SISTEMA OPERACIONAL CP/M EM SEU CP 500 E TORNEO-O COMPATÍVEL COM MAIS DE 3.000 PROGRAMAS DISPONÍVEIS NO MERCADO, POR APENAS:

Cr$ 298.000,00*

REPRESENTANTE DA

Microsol Ltda.

(Fornecedor da Embratel)

INTERSOFT MICROCOMPUTADORES

Av. Brig. Faria Lima, 1.462 - CJ.2 A
Tels.: (011) 211 0371/212 8971
São Paulo S.P.

***PROMOÇÃO VÁLIDA ATÉ TRINTA DE NOVEMBRO DE 1.984, PARA PAGAMENTO À VISTA**

Os seqüenciadores **bit slice** são normalmente de 4 bits. Exemplos representativos são o **ADVANCED MICRO DEVICES 2909 e 2911**, figura 26 e o **TEXAS INSTRUMENTS 74LS482**. Enquanto os seqüenciadores **bit slice** são facilmente expandíveis, os seqüenciadores de endereço fixo necessitam de circuitos adicionais e técnicas de "paginação" para estender a faixa de acesso à memória.

Na categoria de endereço fixo, o INTEL 3001 e MONOLITHIC MEMORIES 67110 podem criar endereços de 9 bits e acessar diretamente

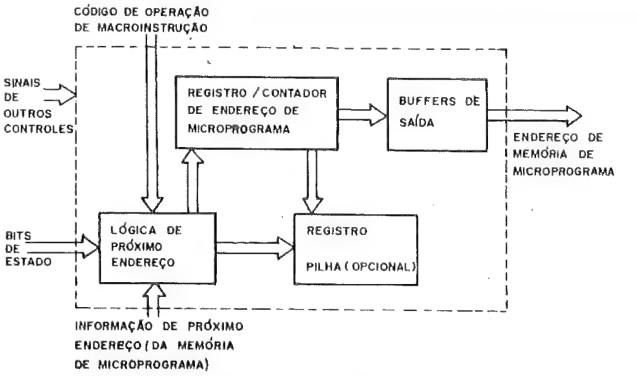


FIGURA 24 – DIAGRAMA EM BLOCO DE UM SEQÜENCIADOR LSI DE MICROPROGRAMA

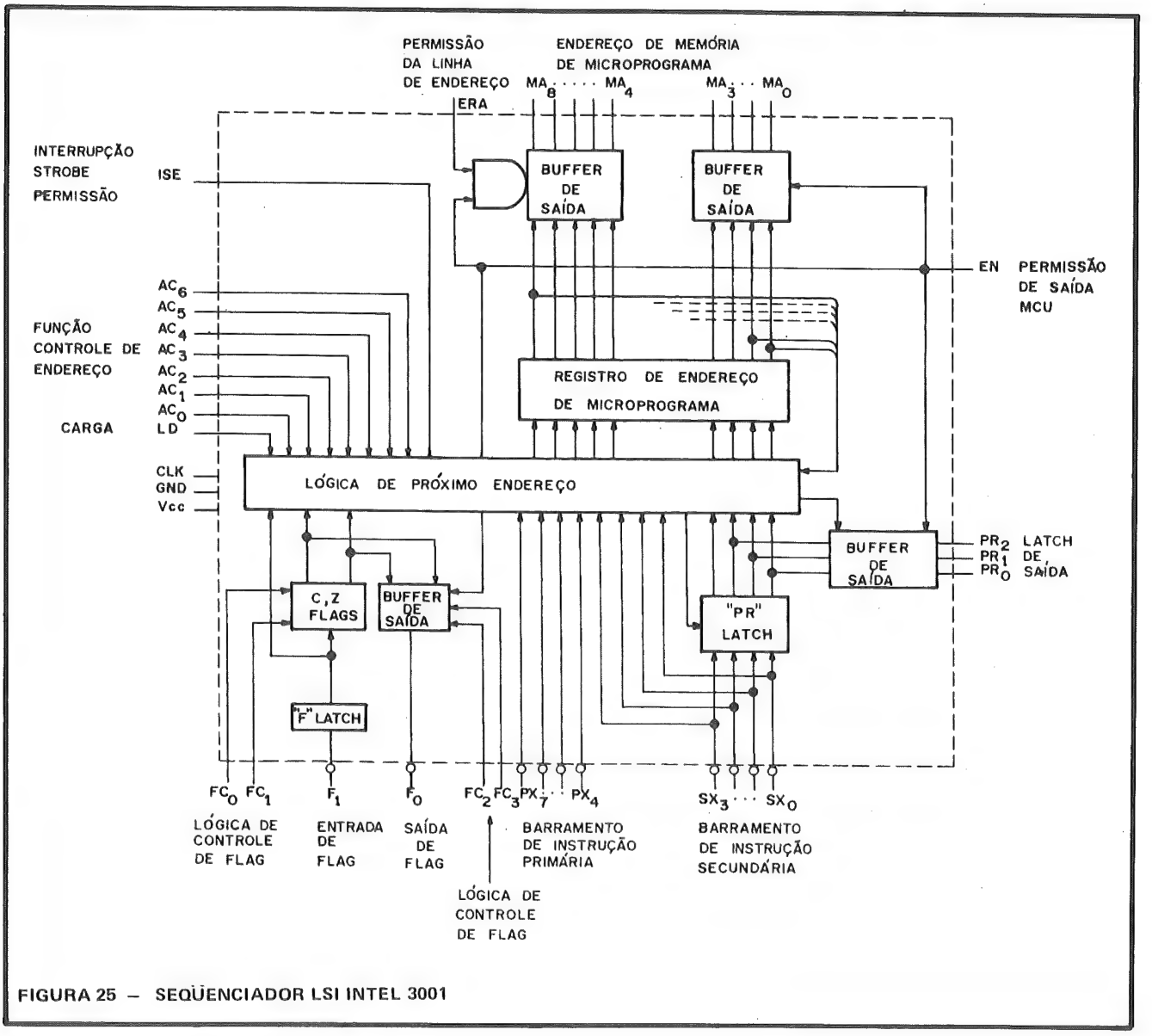


FIGURA 25 – SEQÜENCIADOR LSI INTEL 3001

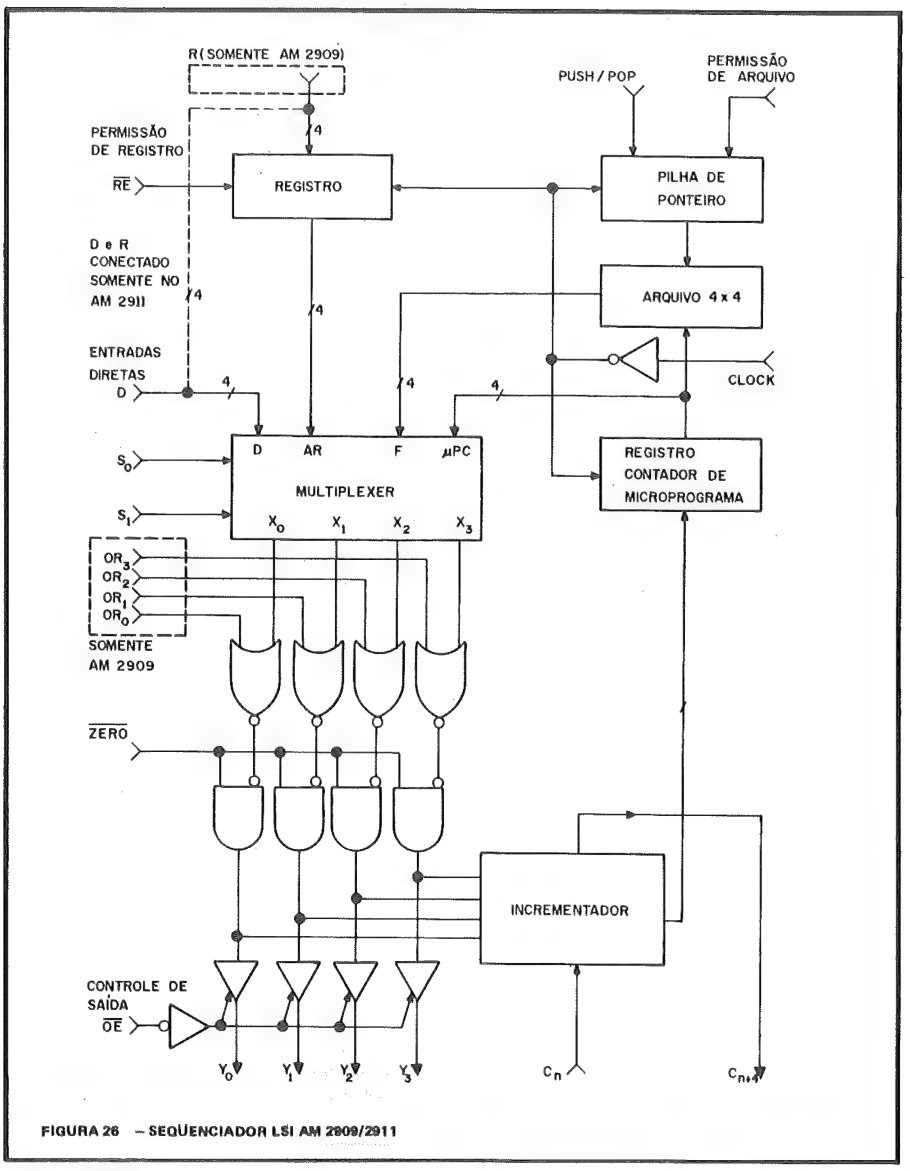


FIGURA 26 – SEQÜENCIADOR LSI AM 2909/2911

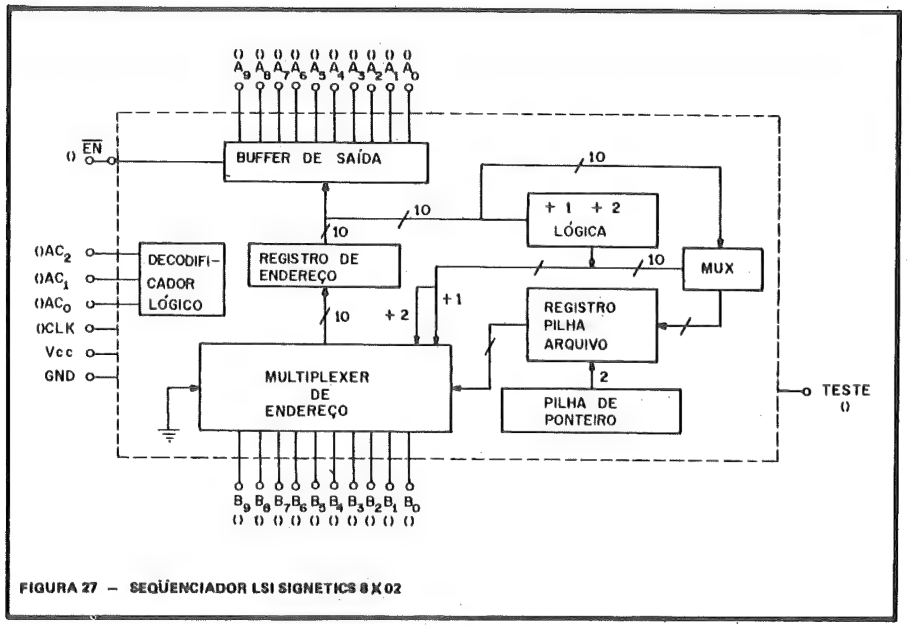


FIGURA 27 – SEQÜENCIADOR LSI SIGNETICS 8 X 02

512 palavras de memória de microprograma, enquanto que o SIGNETICS 8 X 02 e o FAIRCHILD 9408 podem gerar 10 bits de endereço e acessar 1024 palavras da memória de microprograma. Os seqüenciadores **bit slice** de quatro bits (2909/2911, 74S482) são ligados em cascata, o que permite, por exemplo, dois chips endereçarem 256 palavras, três chips endereçarem 4096 palavras e assim sucessivamente. Alguns seqüenciadores novos da **ADVANCED MICRO DEVICES** são o 2910, 12 bits de endereço fixo, e o 2930/2931, **bit slice** de 4 bits.

Essa facilidade de expansão torna-se importante uma vez que as memórias de microprograma e os programas de controle que elas contém estão tornando-se cada vez maiores (vide o caso dos superminicomputadores de 32 bits).

Outra diferença entre os seqüenciadores LSI disponíveis é que vários deles (3001, 8 X 02, 67110, 9408 e 74S482) dispõem no chip de **latches** de saída de endereço, enquanto outros (2909/2911) não possuem os **latches**.

Isto permite maior flexibilidade, porém, requer ao projetista acrescentar seus próprios **latches**. Os **latches** de saída de endereço no chip permitem ao projetista utilizar uma técnica de **PIPELINING** que conduz a uma operação mais rápida, embora isto torne a microprogramação mais difícil.

Alguns dispositivos apresentam um sistema de endereçamento não-convencional. Por exemplo, o seqüenciador INTEL 3001 que trabalha com um espaço de endereço de microprograma organizado numa formação dimensão N. Embora isto possa acarretar uma redução no tamanho da microinstrução e na lógica de próximo endereço, resulta em algumas perdas de capacidade de endereçamento e num aumento de esforço na microprogramação, já que o programador tem que "guardar na mente" a posição exibida no desvio de microprograma.

A tabela 1, apresenta uma lista seqüenciadores e seus principais fabricantes, incluindo ainda um fabricante secundário.

Os tempos de ciclo desses seqüenciadores são diferentes. Por exemplo, o do 8 X 02 é 44ns, do 3001, 2909, 2911 e 74S482 é 50ns e o do 67110 é 100ns.

TABELA 1

SEQÜENCIADORES LSI	FABRICANTE
2909/2911, 2910, 2930/2931	AMD (segunda fonte RAYTHEON, SIGNETICS)
3001 MCU	INTEL (segunda fonte SIGNETICS)
8 x 02 3001 (C.S.S.)	SIGNETICS
6710, 67110 (Controller)	MONOLITHIC MEMORIES
74S482 9408 (Program Controller)	TEXAS INSTRUMENTS FAIRCHILD
10801 (Microprogram Control Device)	MOTOROLA
———	NATIONAL CROM

CONSIDERAÇÕES GERAIS

Como mencionamos anteriormente, a novidade na arquitetura bit slice é a divisão vertical da seção de processamento da CPU em vez da divisão horizontal convencional. Esse tipo de partição vertical em slices divide os registros e a ULA em tamanhos iguais e partes funcionalmente equivalentes, chamadas RALU's ou **bit slices**, figura 28.

Esses bit slices substituem um grande número de dispositivos MSI, freqüentemente utilizados nessas funções. As RALU's podem operar com dois ou quatro bits e podem ser conectadas, em cascata, para formarem uma seção de processamento maior, na qual todas as linhas de controle similares de cada RALU são conectadas em paralelo, sendo que o **carry** de saída de um chip é conectado ao **carry** de entrada do próximo. As operações aritiméticas ou lógicas, as alimentações e o endereçamento para a ULA são os mesmos para todos os **slices**. Em geral, um bloco típico de RALU contém as seguintes seções: **Unidade Lógica e Aritmética** (ULA), múltiplos registros de palavras (1 a 16) com uma ou duas portas (que podem reter operandos e dados temporariamente ou receber os resultados da operação da ULA), um deslocador (shifter), características para E/S de dados (tanto no conjunto de linhas de entrada como de saída, ou múltiplos conjuntos de linhas de entrada multiplexadas internamente), entradas de controle (que podem ser bits de controles de microinstrução decodificados internamente na RALU) e bits de condições de saída (flags).

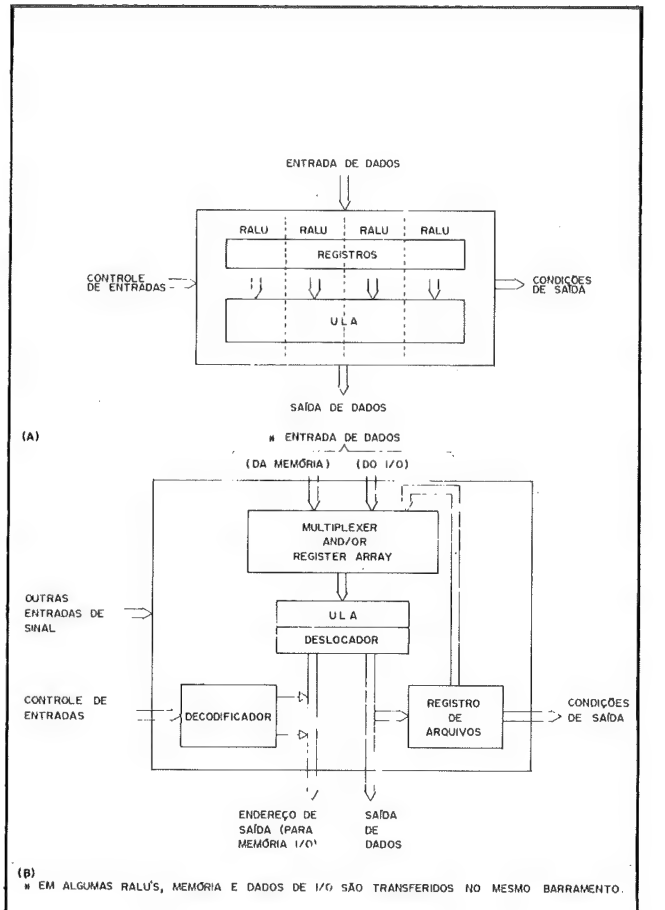


FIGURA 28 – (A) SEÇÃO DE PROCESSAMENTO DIVIDIDA VERTICALMENTE
(B) DIAGRAMA GERAL EM BLOCO DE UMA RALU

A utilização das **RALU's** vertical em vez de porções horizontal de **ULA's** requer menos movimentação de dados, menores microinstruções e menos hardware.

Todas as funções lógicas e aritméticas executadas pelos **bit slices** (RALU's) são fornecidas pela **memória de microprograma** e pelo **seqüenciador de microprograma**. O seqüenciador decodifica o código de operação da macroinstrução "procurada" na memória principal e fornece à memória de microprograma o endereço da microinstrução correspondente à macroinstrução a ser "procurada". Cada microinstrução proveniente da memória de microprograma ocasiona um comando apropriado (sinal de controle) a ser enviado para a seção de processamento do microprocessador bit sliced.

A figura 29 apresenta o diagrama em bloco de dois microprocessadores bit sliced, o INTEL 3002 (2 bits), e o AM 2901 (4 bits). A tabela 2 lista os **microprocessadores bipolares bit sliced** disponíveis no mercado, com os respectivos fabricantes.

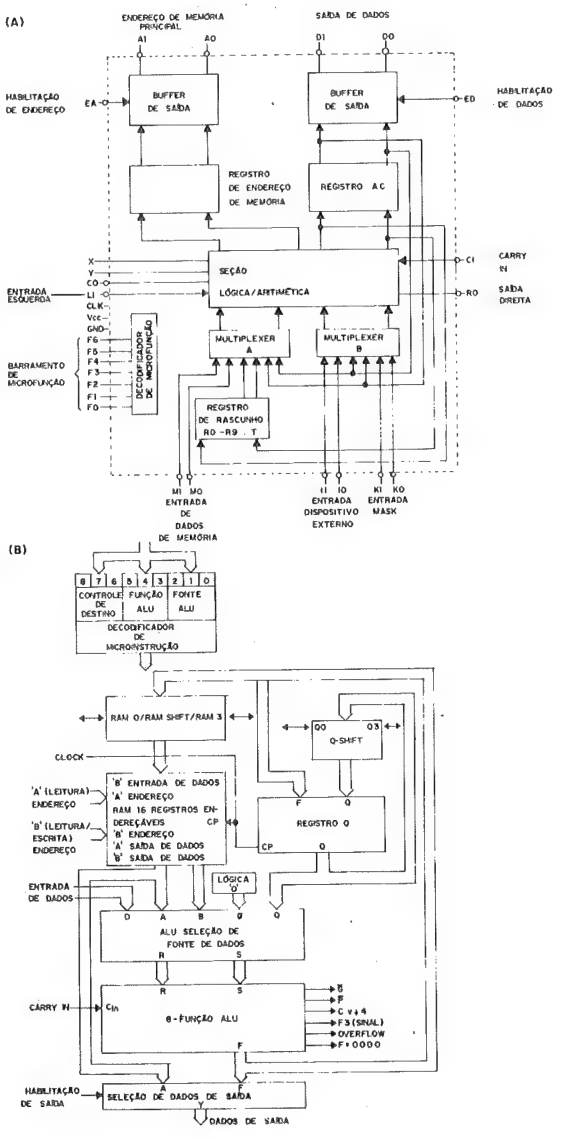


FIGURA 29– DIAGRAMA EM BLOCO DE DOIS MICROPROCESSADORES BIT SLICED DISPONÍVEIS COMERCIALMENTE

(A) O INTEL 3002 (2 bit-RALU)
(B) O AM 2901 (4 bit-RALU)

CONSIDERAÇÕES DE PROJETOS DE SISTEMAS E INTERFACE

Antes de empenhar-se no projeto, o projetista deve definir as especificações e objetivos do sistema.

Qual deve ser o propósito geral do sistema?

Que "liberdades" estão disponíveis para o projetista e para o usuário?

As duas características "chaves" são: velocidade (rendimento ou tempo de resposta) e paralelismo. Ambas influenciam decisivamente na arquitetura do sistema proposto. A primeira determina a tecnologia a ser usada, enquanto a segunda determina a estrutura da memória de microprograma e o formato da microinstrução.

TABELA 2

MICROPROCESSADOR BIT SLICED	FABRICANTE
2901, 2903 (4 bit slice)	AMD (segunda fonte RAYTHEON, MMI, NATIONAL, SIGNETICS)
3002 (2 bit slice)	INTEL (segunda fonte SIGNETICS)
6701 (4 bit slice)	MONOLITHIC MEMORIES
74S481 (4 bit slice) SBP-0400	TEXAS INSTRUMENTS
9405 A (4 bit slice)	FAIRCHILD
10800 (4 bit slice)	MOTOROLA
4057 (4 bit slice)	RCA
GPC/P (4 bit slice)	NATIONAL

Os dispositivos LSI são usados para implementarem antigos softwares em novos hardwares ou para projetar novas e diferentes máquinas.

No primeiro caso o conjunto de macroinstrução está fixado e declarado, enquanto no segundo caso o projetista de sistema tem muito mais flexibilidade e liberdade em definir o conjunto de macroinstrução, melhor adaptável à aplicação específica. Em ambos os casos o projetista, depois de ter definido a macroinstrução mnemônica e especificado seu procedimento de execução, concentra o seu esforço na microarquitetura do sistema para conseguir o melhor e mais eficiente hardware. O seqüenciador de microprograma, a memória de microprograma e a seção de processamento **bit sliced** caracterizam a unidade central de processamento do sistema digital. Em muitos casos a memória prinicpal também é requerida para conservar as macroinstruções de programas aplicativos, igualmente como uma seção de E/S para trocar dados com dispositivos periféricos.

O projetista deve interfacear a memória principal, onde residirão os programas aplicativos, com o resto do sistema. Tal interface exigirá pelo menos um registro de macroendereço (contador de programa para a macromáquina), um registro de macroinstrução, lógica de controle para seleção da memória prinicpal e barramento de **drivers** externo. A seção de processamento **sliced** deve ter acesso ao registro de macroendereço e receber parte do operando da macroinstrução, bem como a palavra completa (ou a palavra de dados) a ser usada, para manipulação de dados e/ou informação. O seqüenciador de microprograma deve receber pelo menos parte do código de operação da macroinstrução, tanto que pode ser interpretado através da seleção da seqüência de microinstrução apropriada. Finalmente, a memória de microprograma deve controlar a memória principal e a eficiência deve ser ganha pela "pré-busca" da macroinstrução, sobrepondo as referências da memória principal com a execução das macroinstruções da seção de processamento.

O modo de E/S é desempenhado (sob controle de pro-

grama de portes, memória mapeada, ou com ajuda de um controle DMA) tendo um efeito sobre toda a arquitetura do sistema e podendo ter um impacto significativo na lógica

Velocidade e paralelismo são duas chaves características que influenciam decisivamente na arquitetura.

necessária. O programa controlador de E/S é mais simples de se fazer com relação ao hardware do que um arranjo mínimo de hardware para implementação da memória mapeada, fazendo com que os periféricos pareçam endereços de memória. A principal desvantagem de ambos é a baixa velocidade, uma vez que gastam tempo de processamento executando transferência de dados. No Acesso Direito à Memória (DMA) o sistema deve pelo menos ser capaz de reconhecer solicitações DMA, parada externa de atividade (HALT) e liberação de controles de barramentos externos (vai para o estado de espera). Se altas velocidades de E/S não são requeridas, a transferência DMA pode ser feita sob controle de microprograma; se altas velocidades são requeridas, chips de controle DMA poderão ser usados.

O modo como as interrupções são manipuladas tem efeito sobre a arquitetura do sistema e requer hardware adicional. O processador interno, uma vez que tenha sido gerada as interrupções (overflow, erros de paridade, etc.), é facilmente manipulado e geralmente necessita de um pequeno serviço de rotina. As interrupções externas geradas pelo usuário têm maior impacto no projeto do sistema e podem ser manipuladas através de **polling ou vetores**. O **polling** é mais fácil de se fazer, porém é mais lento (o microprograma registra cada interrupção para identificar a fonte da interrupção), enquanto que o **vetor** requer menor quantidade de passos de microprograma para implementar, mas necessita de chips controladores especiais.

As interrupções e as sub-rotinas, geradas pelo usuário nos programas de aplicação, têm problemas similares: endereço-de-volta e conservação do estado da máquina.

Em alguns casos, partes da seção de processamento suprem algum tipo de **pilha** para armazenar o endereço-de-volta e conservar parte do estado da máquina, ou dedicam um dos registros internos para um indicador de pilha, apontando para uma seção da memória principal onde o estado da máquina e os registros internos podem ser armazenados. Por outro lado, as micro sub-rotinas usadas no microprograma, geralmente não usam registros da seção de processamento. Freqüentemente, o seqüenciador de microprograma da seção de controle inclui algum tipo de pilha **LIFO** para salvar a **volta-para-o-endereço** das micro sub-rotinas abrigadas.

INSTITUTO DE TECNOLOGIA ORT
CENTRO DE INFORMÁTICA

ORT

CURSOS

LINHA IBM (Apoio Marcodata)
OS/VS1 – VSE – VM/CMS – VSAM
CICS – DL/1 – COBOL: TÉCNICAS E OTIMIZAÇÃO

MICROINFORMÁTICA
BASIC – ASSEMBLER – PASCAL
LOGO – CP/M – VISICALC
dBASE II – WORDSTAR

FORMAÇÃO DE PROGRAMADORES
DURAÇÃO: 9 MESES

CPD-ORT: IBM 4341 COM TERMINAIS
LABORATÓRIO DE MICROS

TREINAMENTO IN HOUSE

SOLICITE INFORMAÇÕES E FOLHETOS EXPLICATIVOS

RUA DONA MARIANA - 213 - BOTAFOGO
TELS.: 226-3192 - 246-9423

Seu micro merece Assistência Técnica ASSIST.

E você merece a tranquilidade de contar com a mais eficiente equipe técnica do Rio, treinada nas fábricas, e recomendada pela Petrobrás, Furnas, Light e Bolsa de Valores. A ASSIST oferece também diversas opções para contratos anuais de assistência técnica, que garantem o máximo ao seu micro.

E sem custar mais por isto.

Os micros Spectrum, Prológica, Digitus e muitos outros, além de vídeo-games e compatibilização de periféricos. têm na ASSIST uma assistência técnica aprovada pelos próprios fabricantes. Além disto, você tem total assistência aos micros importados: Sinclair, TRS-80, Apple e PC/IBM.

Se você tem um micro e quer o máximo em assistência técnica, não pense duas vezes: pense ASSIST.

ASSIST: A máxima solução para seu micro.

ASSIST
Assessoria de Sistemas e Engenharia Ltda.
Av. Beira-Mar, 406 - Gr. 805 - Castelo
Tel.: 262-5763

MICRO WARE
MICRO WARE

CURSOS
INICIAÇÃO AOS MICROCOMPUTADORES
BASIC BÁSICO E AVANÇADO
ASSEMBLER

VENDAS
• MICRO-COMPUTADORES • PROGRAMAS • FITAS • DISQUETES • SUPRIMENTOS • LIVROS • REVISTAS

CONSULTORIA
DESENVOLVIMENTO DE HARDWARE
DESENVOLVIMENTO DE SOFTWARE
ESPECIFICAÇÕES DE SISTEMAS

ASSISTÊNCIA TÉCNICA

EM NITERÓI
Rua Moreira Cesar, 229 Conjunto, 1713
Tel.: (021) 710-2780

Um balanço considerável do desempenho versus a exigência do projeto geralmente requerem vários projetos de interação. No final do processo o projetista deve fazer avaliações cuidadosas das partes consideradas e custos com as subseções, antes de começar a construção do sistema final.

SUPORTE PARA O PROJETO

Apoio para projeto, suporte de software e depuração para máquinas bit sliced não são disponíveis como o suporte para projeto de minicomputadores. Alguns dos poucos disponíveis são comuns para os microprocessadores CMOS e bit slice. Outros são especificamente aplicáveis às máquinas bit sliceds e podem simplificar a microprogramação e a implementação da arquitetura bit slice. Alguns dos mais valiosos apoios para projetos de máquinas bit sliceds são dados a seguir:

Assemblers:

Traduzem progrmas em código de máquina binário. Com micro assemblers o projetista pode especificar as características da memória de microprograma, formatos de microinstrução, códigos de operação de microsinstruções e formatos de saída. Alguns são eficazes para sistema **time sharing** ou numa forma **cross assembler**. Os exemplos são:

- Rapid from Scientific Micro Systems
- Cromis cross – assembler para Intel 3000
- Signetics Microassembler apoia o Intel 3002, o AMD 2901 e o Sgnetics 8 X 02.

Circuitos emuladores:

Simulam outros chips de microcomputador (microprocessador, seqüenciadores de microprogramas, RALU's). Determinam o estado exato do chip em qualquer instante. Encaixam no soquete do circuito que está sendo emulado e fazem o desenvolvimento e depuração em tempo real mais rápido e mais fácil, permitindo a exibição do estado do sistema, modificando e copiando em tempo real. Exemplos:

- *Intel ICE-30 (emula o 3001 MCU)*
- *Intel ICE-80 (emula o 8080 MPU)*

Simuladores ROM:

São essencialmente emuladores para memórias. Permitem o uso das RAM's para depuração de programas em tempo real, de maneira que podem seguir "livres" **(bug free)** até suas posições finais de ROM/PROM. São compatíveis pino a pino com as ROM's/PROM's padrões. Permitem fácil mudança de conteúdo da ROM durante o desenvolvimento e depuração, podendo proporcionar uma cópia fiel da depuração do registro. Os exemplos são:

- *Scientific Micro Systems ROM Simulator*
- *Intel ROM Simulator para 3601/3604 PROM's*

QUANDO UM SISTEMA MICROPROGRAMÁVEL BIT SLICE FAZ SENTIDO

O projetista de sistema que optar por dispositivos **bit sliced LSI bipolar** confronta-se com uma série de problemas

CIBERNE SOFTWARE

apresenta novas fitas com desafios emocionantes para você!

PARA EQUIPAMENTOS COM LOGICA SINCLAIR

1. VALKIRIE
Pilote a nave Valkirie e parta em busca de dez castelos perdidos. (Exclusividade Ciberne, por Divino C.R. Leitão).
E mais: GUERRILHA CÓSMICA e ZOR.

2. MERCADOR DOS SETE MARES
No século XIX você percorre o mundo a bordo de seu navio, em busca de bons negócios.
E mais: CORRIDA MALUCA e PINBALL (Exclusividade Ciberne, por Divino C.R. Leitão).

3. SUBESPAÇO
Implacável caçada espacial. Totalmente gráfico.
E mais: CAVERNAS DE MARTE (Exclusividade Ciberne, por Divino C.R. Leitão) e COMBOIO ESPACIAL.

4. DEFENSOR 3D
Livre nosso planeta de uma invasão alienígena. Fantásticas simulações tri-dimensionais.
E mais: Q'BERT (Exclusividade Ciberne, por Divino C.R. Leitão) e ASSALTO.

5. ROT I - PLUS
• S.O.G. Sistema operacional, com linguagem gráfica. Infinitas opções de uso. Totalmente em código de máquina (Exclusividade Ciberne, por J. Magal).
• MERGE Possibilita a junção de vários programas, uns aos outros.

6. APLIC'I
• COMP-CALC Rápido, eficiente e totalmente em código de máquina. A melhor versão do já famoso Visi-Calc.
• COMP-ARQ Programa gerador de arquivos. Totalmente em código de máquina. Modele fichas e as acesse pelo campo que quiser.
• COMP-TEXTO De fácil manipulação, totalmente em código de máquina.

PARA EQUIPAMENTOS COM LOGICA TRS-80

1. SIMULADOR DE VÔO
Totalmente gráfico e acompanhado de livro de instruções, com diagramas, tabelas etc.
E mais: PINTOR MALUCO e O DESAFIO DA GALINHA.

2. XADREZ
O mais tradicional dos jogos, reeditado em nova e brilhante versão.
E mais: PATRULHA ARMADA e PÂNICO (totalmente sonorizados).

ADQUIRA ESSES LANÇAMENTOS NO SEU REVENDEDOR CIBERNE MAIS PRÓXIMO. E TAMBÉM: Bichos e Cia., Patrulha Galáctica, Aventura e Mistério, Combate, Rot II e Compasete 20 (Ita virgem).

JVA MICROCOMPUTADORES LTDA.
Distribuição e informações:
Av. Graça Aranha, 145 - sobreloja 01
Tel.: (021) 262.6965
Rio de Janeiro - RJ - CEP 20030

SoftKristian®
Revendedores Autorizados

Rio de Janeiro
Seletronix
República do Líbano, 25-A
Rio de Janeiro - RJ
CEP: 20061

Gachel
R: Dr. ElJaick, 25 S/5
Nova Friburgo - RJ
tel.: 22.4208

VGC
Av. Brasil, 10 S/07
Araruama - RJ
CEP: 28970

ENTRELIVROS
Av. Rio Branco, 156 - térreo
Rio de Janeiro - RJ

M.C.S.
Visc. de Pirajá, 303/217
Rio de Janeiro - RJ
tel.: 267.8597

Pernambuco
Eletrônica Isabele
R: Porto Alegre, 112
Caruaru - PE
CEP: 55100

Alagoas
Expoente
Av. Siqueira Campos, 838
Maceió - AL
tel.: (082) 223.3979

São Paulo
Imarés
Av. dos Imarés, 457
São Paulo - SP
tel.: 61.4049 - 61.0946

Fotoleo
R: Boa Vista, 314 - 3º andar
São Paulo - SP
tel.: 35.7131 R/32

Memocards
R: Amador Bueno, 855
Ribeirão Preto - SP
tel.: (016) 636.0586

Fotoptica
Alameda Juruá, 434
São Paulo - SP
tel.: 421.5211

Ritz
R: Frei Caneca, 7
Santos - SP
tel.: 35.1792

Computerland
Av. Angélica, 1996
São Paulo - SP
CEP: 01228

Livraria Poliedro
R: Aurora, 704
São Paulo - SP
tel.: 221.6764

RC Microcomputadores
Av. Estados Unidos, 983
Piracicaba - SP
tel.: 33.7018

Rio Grande do Sul
Advancing
R: Andradas, 1560 galeria Malcon 518 Porto Alegre - RS
tel.: 26.8246

J.H. Santos
Pça. Otavio Rocha, 41
Porto Alegre - RS
CEP: 90000

India Center
R: Floriano Peixoto, 1112 conj. 33/43 Santa Maria - RS
tel.: (055) 221.7120

Geremia Ltda.
Av. Julio de Castilhos, 1872
Caxias do Sul - RS
tel.: 221.1299

Nordemaq
Av. Julio de Castilhos, 3240
Caxias do Sul - RS
tel.: 221.3516

Micromega
R: Julio de Castilhos, 441 - 1º andar Novo Hamburgo - RS
tel.: (0512) 93.4721

Bahia
Oficcina
Shopping Center Itaigara lj40 - 1º piso
Salvador - BA
tel.: (071) 248.6666

Santa Catarina
Supermicro Show
R: dos Ilheus, 10 lj 6
Florianópolis - SC
tel.: 22.8770

Paraná
Computique
Av. Batel, 1750
Curitiba - PR
tel.: 243.1731

Madison
Av. Mal. Deodoro, 311
Curitiba - PR
tel.: 224.3422

Minas Gerais
Computronix
R: Sergipe, 1422
Belo Horizonte - MG
tel.: (031) 225.3305

Eletrorádio
R: Aquiles Loba, 441-A
Belo Horizonte - MG
tel.: (031) 222.8903

Micro Poços
R: Assis Figueiredo, 1072
Poços de Caldas - MG
tel.: (035) 721.1883

Blow-Up
Av. Floriano Peixoto, 396
Uberlândia - MG
tel.: 235.1413 - 235.7359

Brasília
Digitec
SCLN 302 bl.A lj.63
Brasília - DF
tel.: (061) 225.4534

* CREDENCIAMOS NOVOS REVENDEDORES PARA TODO O BRASIL

complexos que devem ser avaliados para cada aplicação.
– O processador difital microprogramável bit sliced encontra uma econômica aplicação primária substituindo o hardware MSI e SSI por lógica microprogramada LSI e, projetando um novo, mais rápido e flexível microprocessador, emulando outros computadores existentes para reaproveitamento de um antigo e testado software operacional.

Algumas dessas vantagens principais são:

Os dispositivos bit sliced bipolar LSI tornam as técnicas de projetos possíveis para o médio usuário, uma vez que não necessitam de muitos gastos de desenvolvimento, pois só as grandes companhias têm recursos para isso.

As partes microprogramáveis do processador bit sliced oferecem ao projetista novas técnicas com as quais pode criar sistemas flexíveis e bem estruturados. Tal implementação bem estruturada testa com mais facilidade e faz a depuração com facilidade. A incorporação de microinstruções especiais fornece equipamentos de teste de operação em ordem e capacidade de recuperação de defeito em tempo real.

As instruções especiais de micro código podem ser implementadas, possibilitando um rápido melhoramento, pois as microinstruções são geralmente mais eficientes do que as macroinstruções.

Certas rotinas críticas (E/S especializados, processo de manipulação de interrupção e rotinas de ponto flutuante) devem ser implementadas como microprogramas. Tais microprogramas podem melhorar a atuação do sistema para uma dada aplicação.

Os sistemas digitais microprogramáveis são mais flexíveis e mais adaptáveis a uma nova aplicação ou para maiores necessidades de uma antiga aplicação. Isto pode ser feito adicionando-se novas microinstruções ou modificando as que lá existem.

As arquiteturas microprogramáveis bit sliced permitem ao projetista determinar o tamanho da palavra (word length) para aplicação específica (14, 24 ou 32 bits).

Os dispositivos bit sliced tornam possíveis um tremendo aumento da velocidade de processamento através do uso da tecnologia bipolar.

O acesso ao bit slice, entretanto, tem algumas desvantagens principais:

Uma unidade de controle microprogramada é geralmente mais lenta do que a unidade de controle por hardware. Algumas vezes esta desvantagem deve ser parcialmente superada adicionando-se mais paralelismo ao sistema.

Muitos microprocessadores bit sliced necessitam de um set de instrução, estabelecida no nível da microinstrução, tornando o trabalho do usuário ainda mais difícil.

Os projetos e software de suporte atualmente disponíveis para processadores bit sliced são um tanto limitados.

Os multichips são menos confiáveis que os microprocessadores chip único por causa do maior número de interconexões.

Apesar dessas desvantagens, a arquitetura dos microprocessadores bit sliced é atrativa sempre que uma aplicação requer a produção de um grande número de dispositivos de alta velocidade.

LIVRARIA SISTEMA LTDA.
ESPECIALIZADA
INFORMÁTICA & ENGENHARIA

CONHEÇA
UMA LIVRARIA
DE DEPARTAMENTOS.

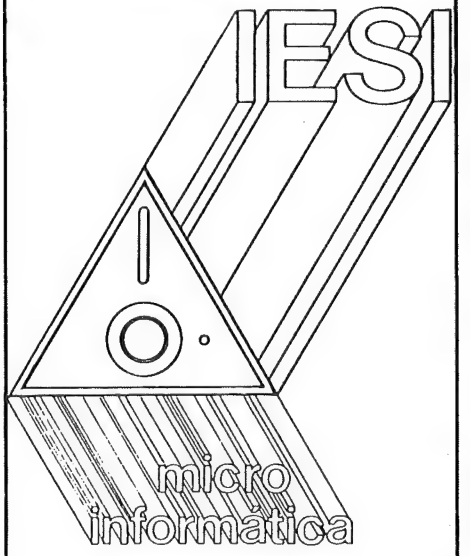
A SISTEMA
tem dentro da Galeria Metropole
duas lojas completas de livros
nacionais e importados.
Sendo uma exclusiva para livros
de INFORMÁTICA COMPUTAÇÃO,
e outra especializada em todas as
áreas da Engenharia, mais
Agropecuária – Veterinária
– Administração-Economia
– Dicionários.

E ainda você pode comprar
em 3 pagtos s/ juros ou ter
um desconto especial
p/ pagamento a vista.
Visite-nos e tenha certeza
de um bom atendimento.

ATENDEMOS POR
REEMBOLSO POSTAL

LIVRARIA SISTEMA
Av. São Luiz, 187 – Sobreloja.
GALERIA METROPOLE
Tels.: 259-1503/257-6118
01046 – Capital – SP

CURSOS DE BASIC
EM "APPLE"
CURSOS DE
APLICATIVOS



VENDAS DE MICROS,
PERIFÉRICOS E SOFTWARE
A PREÇOS ABAIXO DA TABELA
R. Mairinque, 66 — Vl. Mariana
CEP 04037 — Tel.: (011) 572.6988

CP/M no CP-500
MUMPS
DBaseII
COBOL
Pascal UCSD

Estes e mais dezenas de
utilitários estão disponíveis
para seu CP-500 com a
instalação da placa CP/M,
uma exclusividade da

Telematel Informática

Preços de Lançamento:

CP/M ..... Cr$ 295.000
Expansão 64 k ..... Cr$ 250.000
Tela 24x 80 col ..... Cr$ 450.000

$ Promoção Especial$

CP/M + 64 k..... Cr$ 395.000

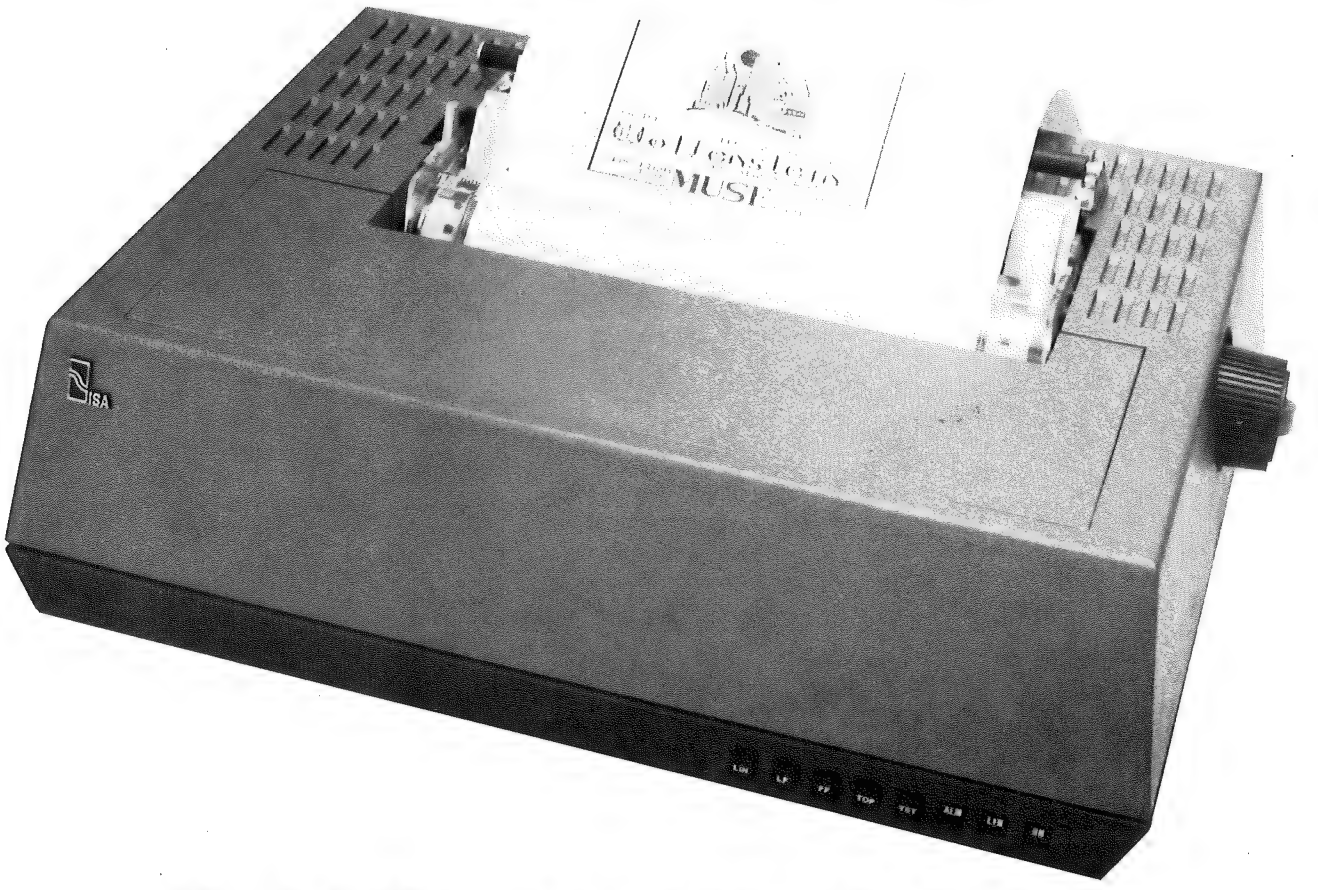
Preços Válidos até 30/10/84

Informações e Vendas:

• Telematel Informática •
Rua Engenheiro Monlevade, 166
São Paulo — SP — CEP 01409
Tel.: (011) 251.3430

Aceitamos representações para todo o Brasil

Impressoras apaixonadas, micros felizes.
O casamento perfeito.



ECODATA EL 8000
Uma impressora fora de série.

EL 8000 é tudo que um micro precisa de uma impressora. De incrível versatilidade a EL 8000 possue:

- Controle por microprocessador.
- Escrita bi-direcional em qualquer circunstância - procura lógica.
- Imprime em diversos idiomas: Inglês, francês, alemão, espanhol etc.
- Desenha, faz gráficos, produz qualquer imagem que um Micro produza em seu vídeo.
- Troca de formato de impressão dentro da mesma linha.
- Velocidade de comunicação de 50 até 9600 BPS.
- Buffer de recepção até 2.000 caracteres.
- Produz diversos tipos de impressão: normal, negrito, expandido, sublinhado, sub e sobre escrito.
- 80 a 132 caracteres por linha.
- Impressão matricial 9 × 7 permitindo até 8 vias numa velocidade de 100 CPS. Em bobina, folha solta e/ou formulário contínuo.
- Etc... etc... etc.

ECODATA
SISTEMA DE PROCESSAMENTO DE DADOS E TELECOMUNICAÇÕES

Rio de Janeiro · R. Republica do Líbano, 61 · 12º and.
Tel.: (021) 221-4677 · Telex (021) 30187
S. Paulo · R. Frei Caneca, 1119 · Tel.: (011) 284-8311 · Telex (011) 22191
Brasília · SCS 02 · Bloco C nº 41 · SL. 01 · Tel.: (061) 225-1588 · Telex (061) 1750
Porto Alegre · Rua Santa Terezinha, 300 · Tel.: (051)232-3564· Telex (051) 2144
Belo Horizonte: Tel.: (031) 222-7809
Goiânia · Salvador

Márkom

EL 58

O teleimpressor Ecodata 100% brasileiro.

O único teleimpressor totalmente nacional, compacto, com teclado ASCII ou BAUDOT.
O Teleimpressor EL 58 vai acompanhar o crescimento de sua empresa por muitos anos.
Totalmente controlado por Software, pode ser modificado em até 8 modelos diferentes, conforme a necessidade de sua empresa.
Robusta, veloz e com excelente escrita, EL 58 foi concebido para ser utilizado em:

- Rede de Telex.
- Rede de Notícias
- Commodities.
- Rede de Processamento de Dados tais como:
  Renpac
  Interdata etc...

Sua concepção rigorosa, permite à empresa dispor de um aparelho com excelente índice técnico e capacidade ilimitada, em nada deixando a desejar às melhores marcas do mercado internacional.

EL 58 tem alta tecnologia e foi fabricado para aguentar grande carga de trabalho.
Com capacidade de impressão caracter a caracter ou bidirecional, o EL 58 imprime em 8 vias.
BUFFER de 2 K caracteres.
Velocidade de recepção até 9.600 BPS.

ECODATA
SISTEMA DE PROCESSAMENTO DE DADOS E TELECOMUNICAÇÕES
Rio de Janeiro · R. Republica do Líbano, 61 · 12º and.
Tel.: (021) 221-4677 · Telex (021) 30187
S. Paulo · R. Frei Caneca, 1119 · Tel.: (011) 284-8311 · Telex (011) 22191
Brasília · SCS 02 · Bloco C nº 41 · SL. 01 · Tel.: (061) 225-1588 · Telex (061) 1750
Porto Alegre · Rua Santa Terezinha, 300 · Tel.:(051)232-3564· Telex (051) 2144
Belo Horizonte: Tel.: (031) 222-7809
Goiânia · Salvador

Márkom

SKILL
INFORMÁTICA

DBASE II *

REVOLUCIONÁRIO STD
(lançamento exclusivo)

O QUE É IMPORTANTE ?

tempo, dinheiro,
suporte, disponibilidade,
ou tudo ?

STD soma

- ECONOMIA DE TEMPO
- NÃO PAGAR ORTN'S
- FACILIDADE DE ACESSO
- RECICLAGEM PERIÓDICA
- COMUNICAÇÃO ACESSÍVEL
- SUPORTE PERMANENTE

SKILL INFORMATICA S C LTDA.

Av. Brigadeiro Faria Lima, 1.664 - 11.º andar - conj. 1.118 - Tels. 813-0954 e 813-9551 - São Paulo - SP CEP - 01452

* DBASE II É UM PRODUTO ASHTON-TATE REPRESENTADO NO BRASIL PELA DATALOGICA

SKILL INFORMÁTICA

STD DBASE II*

STD

SISTEMA DE TREINAMENTO À DISTÂNCIA, TOTALMENTE DESENVOLVIDO NO BRASIL, EM VIDEO K-7 PERMITINDO TER E MANTER O SEU PESSOAL TREINADO E ATUALIZADO

STD DBASE II* É A SOLUÇÃO PARA A COMPREENSÃO E MELHOR UTILIZAÇÃO DESSA IMPORTANTÍSSIMA "FERRAMENTA" DE TRABALHO

SKILL INFORMÁTICA

EMPRESA INTEIRAMENTE NACIONAL, DEDICADA À CONSULTORIA, TREINAMENTO E ASSESSORIA NA ÁREA DA MICRO-INFORMÁTICA

COMO SUPORTE USE NOSSA HOT LINE

SKILL INFORMÁTICA S/C LTDA.

Av. Brigadeiro Faria Lima, 1.664 - 11.° andar - conj. 1.118 - Tels. 813-0954 e 813-9551 - São Paulo - SP CEP - 01452

* DBASE II É UM PRODUTO ASHTON-TATE REPRESENTADO NO BRASIL PELA DATALOGICA

M 1001. O supermicro da Medidata. Preço de micro, performance de mini.

Quando você decide comprar um computador que atenda às necessidades de sua empresa, dois dados fundamentais são levados em conta: a performance e o preço.

Mas quase sempre ao escolher um, você acaba ficando sem o outro. Só que isso mudou.

A Medidata está lançando o supermicro M 1001, que reúne tecnologia de microprocessador com uma performance de mini a um preço de micro.

Multiusuário, o supermicro M 1001 possui até 10 estações de trabalho, permitmdo a operação "On line" e em teleprocessamento, sendo ideal para aplicações administrativas e gerenciais em empresas industriais, comerciais, financeiras ou de serviços.

Apoiado pela experiência acumulada da Medidata em soluções integradas de processamento de dados, o supermicro M 1001 já nasce com uma completa biblioteca de programas aplicativos, comuns a toda a família de computadores compatíveis Medidata, que inclui os minicomputadores M 2001 e M3001 e o micro profissional M301.

Esta família de computadores proporciona uma grande variedade de soluções para pequenas empresas que buscam o seu desenvolvimento e para a descentralização do processamento de grandes e médias empresas.

Supermicro M 1001. Uma solução com a qualidade Medidata, apoiada por uma eficiente rede de Assistência Técnica.

HARDWARE:

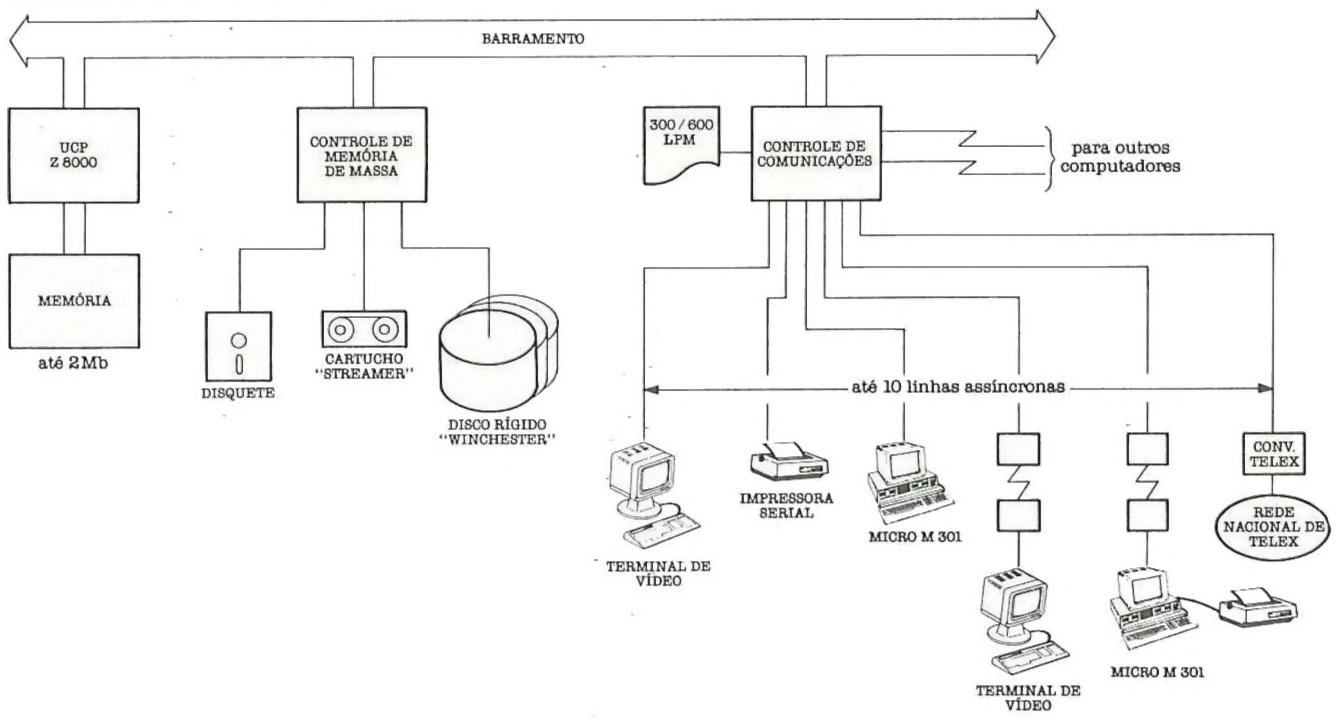
- Micro multiusuário para até 10 terminais, locais ou remotos;
- Memória de até 2 Mb;
- Capacidade de disco de até 45 Mb;
- Impressoras seriais e de linha;
- Cartucho de fita magnética tipo "Streamer", de 25 Mb para "back-up";
- Comunicação síncrona, para ligação com outros computadores.

medidata
a lógica sob medida

R. J.: Rua Rodrigo de Brito, 13 - Tels.: (021) 542-3737 - 286-5552
S. P.: Av. Paulista, 777 - 1.º andar - Tel.: (011) 288-3522
B. H.: Rua Guajajaras, 40 - s/607 - Tels.: (031) 226-5719 - 226-5045

CARACTERÍSTICAS TÉCNICAS

ARQUITETURA DO SUPERMICRO M1001



HARDWARE

O hardware do M1001 representa o "Estado da Arte" da tecnologia de microcomputadores.

CPU baseada no microprocessador ZILOG Z8000 de 16 bits, capaz de manipular, inclusive, dados de 32 bits.

MEMÓRIA até 512 Kb com pastilhas de 64 Kbit ou até 2 Mb com pastilhas de 256 Kbit.

DISQUETES até 2 disquetes 5 1/4", dupla face e dupla densidade, com capacidade individual de 320 Kb.

DISCOS RÍGIDOS até 3 discos de 5 1/4" tipo "Winchester" de 15 Mb formatados (total 45 Mb).

FITA MAGNÉTICA unidade cartucho de fita "Streamer" com capacidade de até 25 Mb para back-up dos discos rígidos.

IMPRESSORA DE LINHA impressão por impacto, empregando banda de aço flexível, linha de 132 posições e velocidade de 300 ou 600 lpm.

COMUNICAÇÃO ASSÍNCRONA até 10 canais de comunicação assíncrona, padrão elétrico RS-232C e velocidades de 75 a 19200 bps, aos quais poderão ser conectados, em modo local ou remoto, qualquer um dos periféricos abaixo:

- Terminal de Vídeo - inteligente, alfanumérico e semigráfico, 25 linhas x 80 colunas, teclado destacável e controle de impressora escrava.
- Impressoras Seriais - impressão matricial com as seguintes opções:
  - 80 colunas, 100 cps, gráfica e qualidade carta (opcionais).
  - 132 colunas, 100 cps, gráfica e qualidade carta.
  - 136 colunas, 250 cps, gráfica.
- Conversor de TELEX - permite a conexão do M1001 à Rede Nacional de Telex.
- Microcomputador M301 - Micro Profissional compatível com o IBM-PC que também oferece compatibilidade com o M1001 (MUMPS) e que pode ser ligado, emulando um terminal de vídeo, permitindo também a transferência de arquivos.

COMUNICAÇÃO SÍNCRONA até dois canais síncronos, padrão RS-232C, para comunicação com outros computadores.

ALIMENTAÇÃO DE EMERGÊNCIA acionada nos casos de queda da rede elétrica, preserva o conteúdo da memória RAM, possibilitando a continuidade da execução dos programas ao se restabelecerem as condições normais de alimentação, o que garante a integridade dos Bancos de Dados utilizados.

SOFTWARE

A MEDIDATA oferece para o M1001 um amplo conjunto de software especialmente voltado para as áreas gerencial, administrativa e financeira de empresas industriais, comerciais ou de serviços, que o tornam o equipamento mais completo de sua classe disponível no mercado brasileiro.

SOFTWARE BÁSICO

MUMPS Sistema Operacional Multiusuário e Multitarefa, Linguagem de Programação e Gerente de Banco de Dados, formando um conjunto perfeitamente integrado, com reconhecida simplicidade e eficiência.

SOFTWARE DE APOIO

GAP Gerência de Aplicações - ferramenta de aumento de produtividade das atividades de análise e programação.

VISTAR/PP Linguagem de "Query" de 4.ª geração para usuários leigos em programação extrairem informações dos Bancos de Dados MUMPS.

SOFTWARE APLICATIVO

PES	Administração de Pessoal	**EST**	Estoque
CBD	Contabilidade Geral	**SAC**	Administração de Compras
CAR	Contas a Receber	**EPM**	Engenharia e Planejamento de Materiais
CAP	Contas a Pagar	**PLANCOM**	Planejamento e Acompanhamento de Projetos
AFX	Ativo Fixo		
FAT	Faturamento		

SINTA NOS DEDOS ESTA CONQUISTA

DIGIPLEX

DIGITUS

DIGITUS

Para pequenas e médias empresas, a DIGITUS lança o DIGIPLEX. Um módulo capaz de formar uma rede local de multi-usuários, que além de proporcionar o dinamismo de um CPD também simplificará o gerenciamento de sua empresa.

Com vários terminais executando programas específicos, a implantação do DIGIPLEX proporcionará a sua empresa um aumento da produtividade e qualidade, já que a interligação on line dos terminais permitirá que se trabalhe com dados e informações atualizadas.

Ligados ao DIGIPLEX poderão estar até 16 terminais inteligentes, fazendo a contabilidade, controle de estoque, vendas e produção, malas diretas, estatísticas ou seja, atendendo a todas as necessidades de sua empresa.

DIGITUS

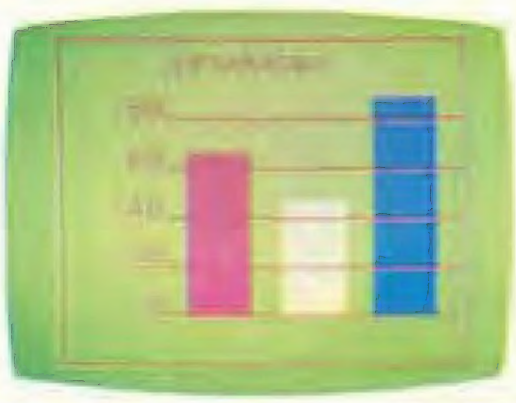
Rua Gávea, 150 - Tel: (031) 332-8300
30.000 - Belo Horizonte - Telex 031-3352
Rua Barata Ribeiro, 391 - s/ 404 -
Tel: (021) 257-2960 - Rio de Janeiro

Revendedores: **Aracajú** (079) 224.7776 223.1310 **Baumeri** (011) 421.5211 **Brasília** (061) 242.6344 248.5359 273.2128 229.4534 **Belém** (091) 225.4000 **Belo Horizonte** (031) 223.6947 222.7889 334.2822 344.5506 225.3305 225.6239 **Campinas** (0192) 32.6322 **Curitiba** (041) 232.1750 243.1731 **Divinópolis** (037) 221.9800 **Fortaleza** (085) 227.5878 224.4235 224.3923 224.4691 226.4922 **Florianópolis** (0482) 23.1039 **Foz do Iguaçú** (0455) 72.1418 **Goiânia** (062) 223.1165 **João Pessoa** (083) 221.6743 **Juiz de Fora** (032) 213.2494 **Londrina** (0432) 23.7110 **Maceió** (082) 223.3979 **Montes Claros** (038) 221.2599 **Niterói** (021) 710.2780 **Novo Hamburgo** (051) 293.1024 **Ouro Preto** (031) 551.3013 **Poços de Caldas** (035) 721.5810 **Porto Alegre** (0512) 26.1988 334.0660 21.4189 25.0007 26.1900 **Recife** (081) 326.9318 221.4995 326.9969 **Ribeirão Preto** (016) 636.0586 **Rio de Janeiro** (021) 252.9420 262.2661 292.0033 267.1093 252.9191 541.2345 268.7480 221.8282 288.2650 253.3395 257.4398 222.4515 263.1241 295.8194 247.7842 322.1960 316.4966 551.8942 **Salvador** (071) 242.9394 241.6189 **Santa Maria** (055) 221.9588 **São Paulo** (011) 280.2322 815.0099 533.2111 231.3922 258.4411 222.1511 853.9288 **Taubaté** (0122) 32.9807 **Vitória** (027) 223.5147 223.5610

CP400

MICROCOMPUTADOR=COLOR

VOCÊ TEM QUE ESTAR PREPARADO PARA SE DESENVOLVER COM OS NOVOS TEMPOS QUE ESTÃO AÍ. E O CP 400 COLOR É A CHAVE DESSA EVOLUÇÃO PESSOAL E PROFISSIONAL.



POR QUÊ? PORQUE O CP 400 COLOR É UM COMPUTADOR PESSOAL DE TEMPO INTEGRAL: ÚTIL PARA A FAMÍLIA TODA, O DIA INTEIRO.

NA HORA DE SE DIVERTIR, POR EXEMPLO, É MUITO MAIS EMOCIONANTE PORQUE, ALÉM DE OFERECER JOGOS INÉDITOS, É O ÚNICO COM 2 JOYSTICKS ANALÓGICOS DE ALTA SENSIBILIDADE, QUE PERMITEM MOVIMENTAR AS IMAGENS EM TODAS AS DIREÇÕES, MESMO. NA HORA DE TRABALHAR E ESTUDAR, O CP 400 COLOR MOSTRA O SEU LADO SÉRIO: MEMÓRIA EXPANSÍVEL, PORTA PARA COMUNICAÇÃO DE DADOS, SAÍDA PARA IMPRESSORA, E UMA ÓTIMA NITIDEZ COM IMAGENS COLORIDAS.

COMO SE TUDO ISSO NÃO BASTASSE, A PROLÓGICA AINDA OFERECE A GARANTIA DE QUALIDADE DE QUEM É LÍDER NA TECNOLOGIA DE COMPUTADORES, E O PREÇO MAIS ACESSÍVEL NA CATEGORIA.

NUMA FRASE: SE VOCÊ NÃO QUISER CHEGAR ATRASADO AO FUTURO, COMPRE SEU CP 400 COLOR IMEDIATAMENTE.

EMOÇÃO E INTELIGÊNCIA NUM EQUIPAMENTO SÓ.

- MICROPROCESSADOR: 6809E COM ESTRUTURA INTERNA DE 16 BITS E CLOCK DE FREQÜÊNCIA DE ATÉ 1.6 MHZ.
- POSSIBILITA O USO DE ATÉ 9 CORES, E TEM UMA RESOLUÇÃO GRÁFICA SUPERIOR A 49.000 PONTOS.
- MEMÓRIA ROM: 16K BYTES PARA SISTEMA OPERACIONAL E INTERPRETADOR BASIC.
- MEMÓRIA RAM: O CP 400 COLOR ESTÁ DISPONÍVEL EM DOIS MODELOS:
  - MODELO 16K: EXPANSÍVEL A 64K BYTES.
  - MODELO 64K: ATÉ 64K BYTES QUANDO USADO COM O NOVO DISK-SYSTEM, CP 450.

- O CP 400 COLOR DISPÕE DE CARTUCHOS DE PROGRAMAS COM 16K BYTES DE CAPACIDADE, QUE PERMITEM O CARREGAMENTO INSTANTÂNEO DE JOGOS, LINGUAGENS E APLICATIVOS COMO: BANCO DE DADOS, PLANILHAS DE CÁLCULO, EDITORES DE TEXTOS, APLICATIVOS FINANCEIROS, APLICATIVOS GRÁFICOS, ETC.
- SAÍDA SERIAL RS 232 C QUE PERMITE COMUNICAÇÃO DE DADOS: ALÉM DO QUE, ATRAVÉS DESTA PORTA, VOCÊ PODE CONECTAR QUALQUER IMPRESSORA SERIAL OU ATÉ MESMO FORMAR UMA REDE DE TRABALHO COM OUTROS MICROS.

- PORTA PARA GRAVADOR CASSETE COM GRAVAÇÃO E LEITURA DE ALTA VELOCIDADE.
- SAÍDAS PARA TV EM CORES E MONITOR PROFISSIONAL.
- DUAS ENTRADAS PARA JOYSTICKS ANALÓGICOS QUE OFERECEM INFINITAS POSIÇÕES NA TELA, ENQUANTO OUTROS TÊM SOMENTE 8 DIREÇÕES.
- AMPLA BIBLIOTECA DE SOFTWARE JÁ DISPONÍVEL.
- ALIMENTAÇÃO: 110-220 VOLTS.

VEJA, TESTE E COMPRE SEU CP 400 COLOR NOS MAGAZINES E REVENDEDORES PROLÓGICA.

TECNOLOGIA PROLOGICA

CP

COMPUTADORES PESSOAIS

RUA PTOLOMEU, 650 - VILA SOCORRO
SÃO PAULO, S.P. - CEP 04762
FONES: (PBX) 523-9933/548-0749/548-4540

QUEM TEM UM, TEM FUTURO.